

---

# Amazon Simple Storage Service

## API Reference

### API Version 2006-03-01



## Amazon Simple Storage Service: API Reference

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Amazon S3 REST API Introduction .....	1
Common Request Headers .....	2
Common Response Headers .....	4
Error Responses .....	6
REST Error Responses .....	6
List of Error Codes .....	7
Authenticating Requests (AWS Signature Version 4) .....	14
Authentication Methods .....	15
Introduction to Signing Requests .....	15
Using an Authorization Header .....	16
Overview .....	16
Signature Calculation: Transfer Payload in a Single Chunk .....	18
Signature Calculation: Transfer Payload in Multiple Chunks .....	29
Using Query Parameters .....	36
Calculating a Signature .....	38
An Example .....	40
Examples: Signature Calculations .....	41
Signature Calculation Examples Using Java .....	41
Signature Calculation Examples Using C# .....	42
Authenticating HTTP POST Requests .....	43
Calculating a Signature .....	44
Amazon S3 Signature Version 4 Authentication Specific Policy Keys .....	45
Bucket Policy Examples Using Signature Version 4 Related Condition Keys .....	47
Browser-Based Uploads Using POST .....	49
Browser-Based Uploads Using HTTP POST .....	49
Calculating a Signature .....	50
Creating HTML Forms .....	51
HTML Form Declaration .....	52
HTML Form Fields .....	52
Creating a POST Policy .....	56
Expiration .....	56
Condition Matching .....	56
Conditions .....	57
Character Escaping .....	59
POST Upload Example .....	61
Uploading a File to Amazon S3 Using HTTP POST .....	61
Using POST with Adobe Flash .....	63
Using POST with Adobe Flash .....	63
Browser-Based Uploads Using AWS Amplify .....	63
Using the AWS Amplify JavaScript library to Upload Files to Amazon S3 .....	64
More Info .....	64
Operations on the Service .....	65
GET Service .....	65
Description .....	65
Requests .....	65
Responses .....	65
Examples .....	67
Related Resources .....	67
Operations on AWS Accounts .....	68
DELETE PublicAccessBlock .....	68
Description .....	68
Requests .....	68
Responses .....	69
Related Resources .....	69

GET PublicAccessBlock .....	69
Description .....	69
Requests .....	69
Responses .....	70
Examples .....	71
Related Resources .....	71
PUT PublicAccessBlock .....	72
Description .....	72
Requests .....	72
Responses .....	74
Examples .....	74
Related Resources .....	75
Operations on Buckets .....	76
DELETE Bucket .....	78
Description .....	78
Requests .....	78
Responses .....	78
Examples .....	78
Related Resources .....	79
DELETE Bucket analytics .....	80
Description .....	80
Requests .....	80
Responses .....	81
Examples .....	81
Related Resources .....	81
DELETE Bucket cors .....	82
Description .....	82
Requests .....	82
Responses .....	82
Examples .....	82
Related Resources .....	83
DELETE Bucket encryption .....	84
Description .....	84
Requests .....	84
Responses .....	84
Examples .....	84
Related Resources .....	85
DELETE Bucket inventory .....	86
Description .....	86
Requests .....	86
Responses .....	87
Examples .....	87
Related Resources .....	87
DELETE Bucket lifecycle .....	88
Description .....	88
Requests .....	88
Responses .....	88
Examples .....	89
Related Resources .....	89
DELETE PublicAccessBlock .....	89
Description .....	89
Requests .....	89
Responses .....	90
Related Resources .....	90
DELETE Bucket metrics .....	90
Description .....	90
Requests .....	91

<b>DELETE Bucket policy</b> .....	93
Description .....	93
Requests .....	93
Responses .....	93
Examples .....	94
Related Resources .....	94
<b>DELETE Bucket replication</b> .....	95
Description .....	95
Requests .....	95
Responses .....	95
Examples .....	95
Related Resources .....	96
<b>DELETE Bucket tagging</b> .....	97
Description .....	97
Requests .....	97
Responses .....	97
Examples .....	97
Related Resources .....	98
<b>DELETE Bucket website</b> .....	99
Description .....	99
Requests .....	99
Responses .....	99
Examples .....	100
Related Resources .....	100
<b>GET Bucket (List Objects) Version 2</b> .....	101
Description .....	101
Requests .....	101
Responses .....	103
Examples .....	106
More Info .....	110
<b>GET Bucket (List Objects) Version 1</b> .....	111
<b>GET Bucket accelerate</b> .....	120
Description .....	120
Requests .....	120
Responses .....	121
Examples .....	121
Related Resources .....	122
<b>GET Bucket acl</b> .....	123
Description .....	123
Requests .....	123
Responses .....	123
Examples .....	125
Related Resources .....	125
<b>GET Bucket analytics</b> .....	126
Description .....	126
Requests .....	126
Responses .....	127
Examples .....	129
Related Resources .....	130
<b>GET Bucket cors</b> .....	131
Description .....	131
Requests .....	131
Responses .....	131
Special Errors .....	133
Examples .....	133
Related Resources .....	134
<b>GET Bucket encryption</b> .....	135

Description .....	135
Requests .....	135
Responses .....	135
Examples .....	137
Related Resources .....	138
<b>GET Bucket Inventory</b> .....	139
Description .....	139
Requests .....	139
Responses .....	140
Examples .....	143
Related Resources .....	144
<b>GET Bucket lifecycle</b> .....	145
Description .....	145
Requests .....	145
Responses .....	145
Special Errors .....	150
Examples .....	150
Related Resources .....	151
<b>GET Bucket location</b> .....	152
Description .....	152
Requests .....	152
<b>GET PublicAccessBlock</b> .....	153
Description .....	153
Requests .....	154
Responses .....	154
Examples .....	155
Related Resources .....	156
<b>GET Bucket logging</b> .....	157
Description .....	157
Requests .....	157
Responses .....	157
Examples .....	159
Related Resources .....	159
<b>GET Bucket metrics</b> .....	160
Description .....	160
Requests .....	160
Responses .....	160
Examples .....	162
Related Resources .....	163
<b>GET Bucket notification</b> .....	164
Description .....	164
Requests .....	164
Responses .....	164
Examples .....	167
Related Resources .....	168
<b>GET Bucket object lock configuration</b> .....	169
Request Syntax .....	169
URI Request Parameters .....	169
Request Body .....	169
Response Syntax .....	169
Response Elements .....	169
Related Resources .....	170
<b>GET BucketPolicyStatus</b> .....	170
Description .....	170
Requests .....	170
Responses .....	170
Examples .....	171

Related Resources .....	171
<b>GET Bucket Object versions</b> .....	173
Description .....	173
Requests .....	173
Responses .....	174
Examples .....	178
Related Resources .....	184
<b>GET Bucket policy</b> .....	185
Description .....	185
Requests .....	185
Responses .....	185
Examples .....	186
Related Resources .....	186
<b>GET Bucket replication</b> .....	187
Description .....	187
Requests .....	187
Responses .....	187
Special Errors .....	192
Examples .....	192
Related Resources .....	193
<b>GET Bucket requestPayment</b> .....	194
Description .....	194
Requests .....	194
Responses .....	194
Examples .....	195
Related Resources .....	195
<b>GET Bucket tagging</b> .....	196
Description .....	196
Requests .....	196
Responses .....	196
Examples .....	197
Related Resources .....	198
<b>GET Bucket versioning</b> .....	199
Description .....	199
Requests .....	199
Responses .....	200
Examples .....	200
Related Resources .....	201
<b>GET Bucket website</b> .....	202
Description .....	202
Requests .....	202
Responses .....	202
Examples .....	203
Related Resources .....	203
<b>HEAD Bucket</b> .....	204
Description .....	204
Requests .....	204
Responses .....	204
Examples .....	205
<b>List Bucket Analytics Configurations</b> .....	206
Description .....	206
Requests .....	206
Responses .....	207
Examples .....	208
Related Resources .....	209
<b>List Bucket Inventory Configurations</b> .....	210
Description .....	210

Requests .....	210
Responses .....	211
Examples .....	212
Related Resources .....	214
List Bucket Metrics Configurations .....	215
Description .....	215
Requests .....	215
Responses .....	216
Examples .....	216
Related Resources .....	217
List Multipart Uploads .....	218
Description .....	218
Requests .....	218
Responses .....	220
Examples .....	223
Related Actions .....	226
PUT Bucket .....	227
Description .....	227
Requests .....	227
Examples .....	230
Related Resources .....	231
PUT Bucket accelerate .....	232
Description .....	232
Requests .....	232
Responses .....	233
Examples .....	233
Related Resources .....	234
PUT Bucket acl .....	235
Description .....	235
Requests .....	235
Responses .....	239
Examples .....	239
Related Resources .....	241
PUT Bucket analytics .....	242
Description .....	242
Requests .....	242
Responses .....	245
Examples .....	246
Related Resources .....	247
PUT Bucket cors .....	248
Description .....	248
Requests .....	249
Responses .....	251
Examples .....	252
Related Resources .....	252
PUT Bucket encryption .....	254
Description .....	254
Requests .....	254
Responses .....	256
Examples .....	256
Related Resources .....	257
PUT Bucket inventory .....	258
Description .....	258
Requests .....	258
Responses .....	262
Examples .....	263
Related Resources .....	264

PUT Bucket lifecycle .....	265
Description .....	265
Requests .....	265
Responses .....	273
Examples .....	274
Related Resources .....	276
PUT PublicAccessBlock .....	277
Description .....	277
Requests .....	277
Responses .....	279
Examples .....	279
Related Resources .....	280
PUT Bucket logging .....	281
Description .....	281
Requests .....	281
Responses .....	284
Examples .....	284
Related Resources .....	285
PUT Bucket metrics .....	285
Description .....	285
Requests .....	285
Responses .....	287
Examples .....	288
Related Resources .....	289
PUT Bucket notification .....	290
Description .....	290
Requests .....	290
Responses .....	294
Examples .....	295
Related Resources .....	297
PUT Bucket object lock configuration .....	298
Request Syntax .....	298
URI Request Parameters .....	298
Request Body .....	298
Response Syntax .....	299
Response Elements .....	299
Related Resources .....	299
PUT Bucket policy .....	300
Description .....	300
Requests .....	300
Responses .....	300
Examples .....	301
Related Resources .....	301
PUT Bucket replication .....	302
Description .....	302
Requests .....	302
Responses .....	309
Examples .....	309
Related Resources .....	310
PUT Bucket requestPayment .....	312
Description .....	312
Requests .....	312
Responses .....	313
Examples .....	313
Related Resources .....	313
PUT Bucket tagging .....	314
Description .....	314

Requests .....	314
Responses .....	315
Examples .....	315
Related Resources .....	316
PUT Bucket versioning .....	317
Description .....	317
Requests .....	317
Responses .....	319
Examples .....	319
Related Resources .....	320
PUT Bucket website .....	321
Description .....	321
Requests .....	321
Responses .....	325
Examples .....	326
DefaultRetention .....	330
Contents .....	330
ObjectLockConfiguration .....	331
Contents .....	331
ObjectLockRule .....	332
Contents .....	332
Operations on Objects .....	333
Delete Multiple Objects .....	333
Description .....	333
Requests .....	334
Responses .....	336
Examples .....	338
Related Actions .....	342
DELETE Object .....	343
Description .....	343
Requests .....	343
Responses .....	344
Examples .....	344
Related Resources .....	346
DELETE Object tagging .....	347
Description .....	347
Requests .....	347
Responses .....	347
Examples .....	347
Related Resources .....	348
GET Object .....	349
Description .....	349
Versioning .....	350
Requests .....	350
Responses .....	354
Examples .....	356
Related Resources .....	360
GET Object ACL .....	361
Description .....	361
Versioning .....	361
Requests .....	361
Responses .....	361
Examples .....	363
Related Resources .....	364
GET Object legal hold .....	365
Request Syntax .....	365
URI Request Parameters .....	365

Request Body .....	365
Response Syntax .....	365
Response Elements .....	365
Related Resources .....	365
GET Object retention .....	366
Request Syntax .....	366
URI Request Parameters .....	366
Request Body .....	366
Response Syntax .....	366
Response Elements .....	366
Related Resources .....	367
GET Object tagging .....	368
Description .....	368
Requests .....	368
Responses .....	368
Examples .....	369
Related Resources .....	370
GET Object torrent .....	371
Description .....	371
Requests .....	371
Responses .....	371
Examples .....	372
Related Resources .....	372
HEAD Object .....	373
Description .....	373
Versioning .....	373
Requests .....	373
Responses .....	376
Examples .....	379
Sample Request for an Glacier Object .....	381
Sample Response - Glacier Object .....	381
Related Resources .....	381
OPTIONS object .....	382
Description .....	382
Requests .....	382
Responses .....	383
Examples .....	384
Related Resources .....	384
POST Object .....	385
Description .....	385
Versioning .....	385
Requests .....	385
Examples .....	395
Related Resources .....	395
POST Object restore .....	397
Description .....	397
Querying Archives with Select Requests .....	397
Restoring Archives .....	398
Requests .....	399
Responses .....	408
Examples .....	409
More Info .....	411
PUT Object .....	412
Description .....	412
Versioning .....	412
Storage Class Options .....	412
Access Permissions .....	412

Requests .....	413
Responses .....	421
Examples .....	422
Related Resources .....	425
PUT Object legal hold .....	427
Request Syntax .....	427
URI Request Parameters .....	427
Request Body .....	427
Response Syntax .....	427
Response Elements .....	427
Related Resources .....	428
PUT Object retention .....	429
Request Syntax .....	429
URI Request Parameters .....	429
Request Body .....	429
Response Syntax .....	430
Response Elements .....	430
Related Resources .....	430
PUT Object - Copy .....	431
Description .....	431
Versioning .....	432
Access Permissions .....	432
Requests .....	432
Responses .....	442
Examples .....	444
Related Resources .....	446
PUT Object acl .....	447
Description .....	447
Versioning .....	447
Requests .....	447
Responses .....	451
Examples .....	452
Related Resources .....	453
PUT Object tagging .....	454
Description .....	454
Requests .....	454
Responses .....	455
Examples .....	456
Related Resources .....	456
SELECT Object Content .....	457
Description .....	457
Requests .....	457
Responses .....	465
Examples .....	481
Notes .....	483
Related Resources .....	483
Abort Multipart Upload .....	484
Description .....	484
Requests .....	484
Responses .....	484
Examples .....	485
Related Actions .....	485
Complete Multipart Upload .....	486
Description .....	486
Requests .....	486
Responses .....	487
Examples .....	489

Related Actions .....	491
Initiate Multipart Upload .....	492
Description .....	492
Requests .....	492
Responses .....	498
Examples .....	500
Related Actions .....	501
List Parts .....	502
Description .....	502
Requests .....	502
Responses .....	503
Examples .....	506
Related Actions .....	507
Upload Part .....	508
Description .....	508
Requests .....	508
Responses .....	510
Examples .....	511
Related Actions .....	512
Upload Part - Copy .....	514
Description .....	514
Requests .....	514
Versioning .....	518
Responses .....	518
Examples .....	520
Related Actions .....	521
ObjectLockLegalHold .....	522
Contents .....	522
ObjectLockRetention .....	523
Contents .....	523
Resources .....	524
Document History .....	525
Appendix .....	541
Appendix: SOAP API .....	541
Operations on the Service (SOAP API) .....	541
Operations on Buckets (SOAP API) .....	542
Operations on Objects (SOAP API) .....	551
SOAP Error Responses .....	566
Appendix: Lifecycle Configuration APIs (Deprecated) .....	568
PUT Bucket lifecycle (Deprecated) .....	569
GET Bucket lifecycle (Deprecated) .....	579
Glossary .....	586

# Amazon S3 REST API Introduction

Welcome to the *Amazon Simple Storage Service API Reference*. This guide explains the Amazon Simple Storage Service (Amazon S3) application programming interface (API). It describes various API operations, related request and response structures, and error codes. The current version of the Amazon S3 API is 2006-03-01.

Amazon S3 supports the REST API.

## Note

Support for SOAP over HTTP is deprecated, but it is still available over HTTPS. However, new Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

Read the following about authentication and access control before going to specific API topics.

Requests to Amazon S3 can be authenticated or anonymous. Authenticated access requires credentials that AWS can use to authenticate your requests. When making REST API calls directly from your code, you create a signature using valid credentials and include the signature in your request. For information about various authentication methods and signature calculations, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

Making REST API calls directly from your code can be cumbersome. It requires you to write the necessary code to calculate a valid signature to authenticate your requests. We recommend the following alternatives instead:

- Use the AWS SDKs to send your requests (see [Sample Code and Libraries](#)). With this option, you don't need to write code to calculate a signature for request authentication because the SDK clients authenticate your requests by using access keys that you provide. Unless you have a good reason not to, you should always use the AWS SDKs.
- Use the AWS CLI to make Amazon S3 API calls. For information about setting up the AWS CLI and example Amazon S3 commands see the following topics:

[Set Up the AWS CLI in the Amazon Simple Storage Service Developer Guide](#).

[Using Amazon S3 with the AWS Command Line Interface](#) in the [AWS Command Line Interface User Guide](#).

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access Amazon S3 resources. For example, you must have permissions to create an S3 bucket or get an object from your bucket. If you use root credentials of your AWS account, you have all the permissions. However, using root credentials is not recommended. Instead, we recommend that you create IAM users in your account and manage user permissions. For more information, see [Managing Access Permissions to Your Amazon S3 Resources](#) in the [Amazon Simple Storage Service Developer Guide](#).

# Common Request Headers

The following table describes headers that can be used by various types of Amazon S3 REST requests.

Header Name	Description
Authorization	The information required for request authentication. For more information, go to <a href="#">The Authentication Header</a> in the <i>Amazon Simple Storage Service Developer Guide</i> . For anonymous requests this header is not required.
Content-Length	Length of the message (without the headers) according to RFC 2616. This header is required for PUTs and operations that load XML, such as logging and ACLs.
Content-Type	The content type of the resource in case the request content in the body. Example: <code>text/plain</code>
Content-MD5	The base64 encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, go to <a href="#">REST Authentication</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .
Date	The current date and time according to the requester. Example: <code>Wed, 01 Mar 2006 12:00:00 GMT</code> . When you specify the Authorization header, you must specify either the <code>x-amz-date</code> or the <code>Date</code> header.
Expect	When your application uses 100-continue, it does not send the request body until it receives an acknowledgment. If the message is rejected based on the headers, the body of the message is not sent. This header can be used only if you are sending a body.  Valid Values: 100-continue
Host	For path-style requests, the value is <code>s3.amazonaws.com</code> . For virtual-style requests, the value is <code>BucketName.s3.amazonaws.com</code> . For more information, go to <a href="#">Virtual Hosting</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  This header is required for HTTP 1.1 (most toolkits add this header automatically); optional for HTTP/1.0 requests.
<code>x-amz-content-sha256</code>	When using signature version 4 to authenticate request, this header provides a hash of the request payload. For more information see <a href="#">Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4)</a> (p. 18). When uploading object in chunks, you set the value to <code>STREAMING-AWS4-HMAC-SHA256-PAYLOAD</code> to indicate that the signature covers only headers and that there is

Header Name	Description
	<p>no payload. For more information, see <a href="#">Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4) (p. 29)</a>.</p>
x-amz-date	<p>The current date and time according to the requester. Example: <code>Wed, 01 Mar 2006 12:00:00 GMT</code>. When you specify the Authorization header, you must specify either the <code>x-amz-date</code> or the <code>Date</code> header. If you specify both, the value specified for the <code>x-amz-date</code> header takes precedence.</p>
x-amz-security-token	<p>This header can be used in the following scenarios:</p> <ul style="list-style-type: none"><li>Provide security tokens for Amazon DevPay operations - Each request that uses Amazon DevPay requires two <code>x-amz-security-token</code> headers: one for the product token and one for the user token. When Amazon S3 receives an authenticated request, it compares the computed signature with the provided signature. Improperly formatted multi-value headers used to calculate a signature can cause authentication issues.</li><li>Provide security token when using temporary security credentials - When making requests using temporary security credentials you obtained from IAM you must provide a security token using this header. To learn more about temporary security credentials, go to <a href="#">Making Requests</a>.</li></ul> <p>This header is required for requests that use Amazon DevPay and requests that are signed using temporary security credentials.</p>

# Common Response Headers

The following table describes response headers that are common to most AWS S3 responses.

Name	Description
Content-Length	<p>The length in bytes of the body in the response.</p> <p>Type: String</p> <p>Default: None</p>
Content-Type	<p>The MIME type of the content. For example, Content-Type: text/html; charset=utf-8</p> <p>Type: String</p> <p>Default: None</p>
Connection	<p>specifies whether the connection to the server is open or closed.</p> <p>Type: Enum</p> <p>Valid Values: open   close</p> <p>Default: None</p>
Date	<p>The date and time Amazon S3 responded, for example, Wed, 01 Mar 2006 12:00:00 GMT.</p> <p>Type: String</p> <p>Default: None</p>
ETag	<p>The entity tag is a hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. The ETag may or may not be an MD5 digest of the object data. Whether or not it is depends on how the object was created and how it is encrypted as described below:</p> <ul style="list-style-type: none"><li>Objects created by the PUT Object, POST Object, or Copy operation, or through the AWS Management Console, and are encrypted by SSE-S3 or plaintext, have ETags that are an MD5 digest of their object data.</li><li>Objects created by the PUT Object, POST Object, or Copy operation, or through the AWS Management Console, and are encrypted by SSE-C or SSE-KMS, have ETags that are not an MD5 digest of their object data.</li><li>If an object is created by either the Multipart Upload or Part Copy operation, the ETag is not an MD5 digest, regardless of the method of encryption.</li></ul> <p>Type: String</p>
Server	<p>The name of the server that created the response.</p> <p>Type: String</p> <p>Default: AmazonS3</p>

Name	Description
x-amz-delete-marker	<p>Specifies whether the object returned was (true) or was not (false) a delete marker.</p> <p>Type: Boolean</p> <p>Valid Values: true   false</p> <p>Default: false</p>
x-amz-id-2	<p>A special token that is used together with the x-amz-request-id header to help AWS troubleshoot problems. For information about AWS support using these request IDs, see <a href="#">Troubleshooting Amazon S3</a>.</p> <p>Type: String</p> <p>Default: None</p>
x-amz-request-id	<p>A value created by Amazon S3 that uniquely identifies the request. This value is used together with the x-amz-id-2 header to help AWS troubleshoot problems. For information about AWS support using these request IDs, see <a href="#">Troubleshooting Amazon S3</a>.</p> <p>Type: String</p> <p>Default: None</p>
x-amz-version-id	<p>The version of the object. When you enable versioning, Amazon S3 generates a random number for objects added to a bucket. The value is UTF-8 encoded and URL ready. When you PUT an object in a bucket where versioning has been suspended, the version ID is always null.</p> <p>Type: String</p> <p>Valid Values: null   any URL-ready, UTF-8 encoded string</p> <p>Default: null</p>

# Error Responses

This section provides reference information about Amazon S3 errors.

**Note**

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

**Topics**

- [REST Error Responses \(p. 6\)](#)
- [List of Error Codes \(p. 7\)](#)

## REST Error Responses

When an error occurs, the header information contains the following:

- Content-Type: application/xml
- An appropriate 3xx, 4xx, or 5xx HTTP status code

The body or the response also contains information about the error. The following sample error response shows the structure of response elements common to all REST error responses.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
  <Resource>/mybucket/myfoto.jpg</Resource>
  <RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

The following table explains the REST error response elements.

Name	Description
Code	The error code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see <a href="#">List of Error Codes (p. 7)</a> .  Type: String  Ancestor: Error
Error	Container for all error elements.  Type: Container  Ancestor: None
Message	The error message contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care

Name	Description
	to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the error message.  Type: String  Ancestor: <code>Error</code>
<code>RequestId</code>	ID of the request associated with the error.  Type: String  Ancestor: <code>Error</code>
<code>Resource</code>	The bucket or object that is involved in the error.  Type: String  Ancestor: <code>Error</code>

Many error responses contain additional structured data meant to be read and understood by a developer diagnosing programming errors. For example, if you send a Content-MD5 header with a REST PUT request that doesn't match the digest calculated on the server, you receive a `BadDigest` error. The error response also includes as detail elements the digest we calculated, and the digest you told us to expect. During development, you can use this information to diagnose the error. In production, a well-behaved program might include this information in its error log.

For information about general response elements, go to [Error Responses](#).

## List of Error Codes

The following table lists Amazon S3 error codes.

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
<code>AccessDenied</code>	Access Denied	403 Forbidden	Client
<code>AccountProblem</code>	There is a problem with your AWS account that prevents the operation from completing successfully. Please contact AWS Support for further assistance, see <a href="#">Contact Us</a> .	403 Forbidden	Client
<code>AllAccessDisabled</code>	All access to this Amazon S3 resource has been disabled. Please contact AWS Support for further assistance, see <a href="#">Contact Us</a> .	403 Forbidden	Client
<code>AmbiguousGrantByEmailAddress</code>	The email address you provided is associated with more than one account.	400 Bad Request	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
AuthorizationHeaderMalformed	The authorization header you provided is invalid.	400 Bad Request	N/A
BadDigest	The Content-MD5 you specified did not match what we received.	400 Bad Request	Client
BucketAlreadyExists	The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.	409 Conflict	Client
BucketAlreadyOwnedByYou	The bucket you tried to create already exists, and you own it. Amazon S3 returns this error in all AWS Regions except us-east-1 (N. Virginia). For legacy compatibility, if you re-create an existing bucket that you already own in us-east-1, Amazon S3 returns 200 OK and resets the bucket access control lists (ACLs).	409 Conflict (in all regions except us-east-1)	Client
BucketNotEmpty	The bucket you tried to delete is not empty.	409 Conflict	Client
CredentialsNotSupported	This request does not support credentials.	400 Bad Request	Client
CrossLocationLoggingProhibited	Cross-location logging not allowed. Buckets in one geographic location cannot log information to a bucket in another location.	403 Forbidden	Client
EntityTooSmall	Your proposed upload is smaller than the minimum allowed object size.	400 Bad Request	Client
EntityTooLarge	Your proposed upload exceeds the maximum allowed object size.	400 Bad Request	Client
ExpiredToken	The provided token has expired.	400 Bad Request	Client
IllegalVersioningConfigurationException	Indicates that the versioning configuration specified in the request is invalid.	400 Bad Request	Client
IncompleteBody	You did not provide the number of bytes specified by the Content-Length HTTP header	400 Bad Request	Client
IncorrectNumberOfFilesInPostRequest	POST requires exactly one file upload per request.	400 Bad Request	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
InlineDataTooLarge	Inline data exceeds the maximum allowed size.	400 Bad Request	Client
InternalError	We encountered an internal error. Please try again.	500 Internal Server Error	Server
InvalidAccessKeyId	The AWS access key ID you provided does not exist in our records.	403 Forbidden	Client
InvalidAddressingHeader	You must specify the Anonymous role.	N/A	Client
InvalidArgument	Invalid Argument	400 Bad Request	Client
InvalidBucketName	The specified bucket is not valid.	400 Bad Request	Client
InvalidBucketState	The request is not valid with the current state of the bucket.	409 Conflict	Client
InvalidDigest	The Content-MD5 you specified is not valid.	400 Bad Request	Client
InvalidEncryptionAlgorithmError	The encryption request you specified is not valid. The valid value is AES256.	400 Bad Request	Client
InvalidLocationConstraint	The specified location constraint is not valid. For more information about Regions, see <a href="#">How to Select a Region for Your Buckets</a> .	400 Bad Request	Client
InvalidObjectState	The operation is not valid for the current state of the object.	403 Forbidden	Client
InvalidPart	One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.	400 Bad Request	Client
InvalidPartOrder	The list of parts was not in ascending order. Parts list must be specified in order by part number.	400 Bad Request	Client
InvalidPayer	All access to this object has been disabled. Please contact AWS Support for further assistance, see <a href="#">Contact Us</a> .	403 Forbidden	Client
InvalidPolicyDocument	The content of the form does not meet the conditions specified in the policy document.	400 Bad Request	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
InvalidRange	The requested range cannot be satisfied.	416 Requested Range Not Satisfiable	Client
InvalidRequest	Please use AWS4-HMAC-SHA256.	400 Bad Request	N/A
InvalidRequest	SOAP requests must be made over an HTTPS connection.	400 Bad Request	Client
InvalidRequest	Amazon S3 Transfer Acceleration is not supported for buckets with non-DNS compliant names.	400 Bad Request	N/A
InvalidRequest	Amazon S3 Transfer Acceleration is not supported for buckets with periods (.) in their names.	400 Bad Request	N/A
InvalidRequest	Amazon S3 Transfer Accelerate endpoint only supports virtual style requests.	400 Bad Request	N/A
InvalidRequest	Amazon S3 Transfer Accelerate is not configured on this bucket.	400 Bad Request	N/A
InvalidRequest	Amazon S3 Transfer Accelerate is disabled on this bucket.	400 Bad Request	N/A
InvalidRequest	Amazon S3 Transfer Acceleration is not supported on this bucket. Contact AWS Support for more information.	400 Bad Request	N/A
InvalidRequest	Amazon S3 Transfer Acceleration cannot be enabled on this bucket. Contact AWS Support for more information.	400 Bad Request	N/A
InvalidSecurity	The provided security credentials are not valid.	403 Forbidden	Client
InvalidSOAPRequest	The SOAP request body is invalid.	400 Bad Request	Client
InvalidStorageClass	The storage class you specified is not valid.	400 Bad Request	Client
InvalidTargetBucketForLogging	The target bucket for logging does not exist, is not owned by you, or does not have the appropriate grants for the log-delivery group.	400 Bad Request	Client
InvalidToken	The provided token is malformed or otherwise invalid.	400 Bad Request	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
InvalidURI	Couldn't parse the specified URI.	400 Bad Request	Client
KeyTooLongError	Your key is too long.	400 Bad Request	Client
MalformedACLError	The XML you provided was not well-formed or did not validate against our published schema.	400 Bad Request	Client
MalformedPOSTRequest	The body of your POST request is not well-formed multipart/form-data.	400 Bad Request	Client
MalformedXML	This happens when the user sends malformed XML (XML that doesn't conform to the published XSD) for the configuration. The error message is, "The XML you provided was not well-formed or did not validate against our published schema."	400 Bad Request	Client
MaxMessageLengthExceeded	Your request was too big.	400 Bad Request	Client
MaxPostPreDataLengthExceededError	Your POST request fields preceding the upload file were too large.	400 Bad Request	Client
MetadataTooLarge	Your metadata headers exceed the maximum allowed metadata size.	400 Bad Request	Client
MethodNotAllowed	The specified method is not allowed against this resource.	405 Method Not Allowed	Client
MissingAttachment	A SOAP attachment was expected, but none were found.	N/A	Client
MissingContentLength	You must provide the Content-Length HTTP header.	411 Length Required	Client
MissingRequestBodyError	This happens when the user sends an empty XML document as a request. The error message is, "Request body is empty."	400 Bad Request	Client
MissingSecurityElement	The SOAP 1.1 request is missing a security element.	400 Bad Request	Client
MissingSecurityHeader	Your request is missing a required header.	400 Bad Request	Client
NoLoggingStatusForKey	There is no such thing as a logging status subresource for a key.	400 Bad Request	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
NoSuchBucket	The specified bucket does not exist.	404 Not Found	Client
NoSuchBucketPolicy	The specified bucket does not have a bucket policy.	404 Not Found	Client
NoSuchKey	The specified key does not exist.	404 Not Found	Client
NoSuchLifecycleConfiguration	The lifecycle configuration does not exist.	404 Not Found	Client
NoSuchUpload	The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.	404 Not Found	Client
NoSuchVersion	Indicates that the version ID specified in the request does not match an existing version.	404 Not Found	Client
NotImplemented	A header you provided implies functionality that is not implemented.	501 Not Implemented	Server
NotSignedUp	Your account is not signed up for the Amazon S3 service. You must sign up before you can use Amazon S3. You can sign up at the following URL: <a href="https://aws.amazon.com/s3">https://aws.amazon.com/s3</a>	403 Forbidden	Client
OperationAborted	A conflicting conditional operation is currently in progress against this resource. Try again.	409 Conflict	Client
PermanentRedirect	The bucket you are attempting to access must be addressed using the specified endpoint. Send all future requests to this endpoint.	301 Moved Permanently	Client
PreconditionFailed	At least one of the preconditions you specified did not hold.	412 Precondition Failed	Client
Redirect	Temporary redirect.	307 Moved Temporarily	Client
RestoreAlreadyInProgress	Object restore is already in progress.	409 Conflict	Client
RequestIsNotMultiPartContent	Bucket POST must be of the enclosure-type multipart/form-data.	400 Bad Request	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
RequestTimeout	Your socket connection to the server was not read from or written to within the timeout period.	400 Bad Request	Client
RequestTimeTooSkewed	The difference between the request time and the server's time is too large.	403 Forbidden	Client
RequestTorrentOfBucketError	Requesting the torrent file of a bucket is not permitted.	400 Bad Request	Client
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your AWS secret access key and signing method. For more information, see <a href="#">REST Authentication</a> and <a href="#">SOAP Authentication</a> for details.	403 Forbidden	Client
ServiceUnavailable	Reduce your request rate.	503 Service Unavailable	Server
SlowDown	Reduce your request rate.	503 Slow Down	Server
TemporaryRedirect	You are being redirected to the bucket while DNS updates.	307 Moved Temporarily	Client
TokenRefreshRequired	The provided token must be refreshed.	400 Bad Request	Client
TooManyBuckets	You have attempted to create more buckets than allowed.	400 Bad Request	Client
UnexpectedContent	This request does not support content.	400 Bad Request	Client
UnresolvableGrantByEmailAddress	The email address you provided does not match any account on record.	400 Bad Request	Client
UserKeyMustBeSpecified	The bucket POST must contain the specified field name. If it is specified, check the order of the fields.	400 Bad Request	Client

# Authenticating Requests (AWS Signature Version 4)

## Topics

- [Authentication Methods \(p. 15\)](#)
- [Introduction to Signing Requests \(p. 15\)](#)
- [Authenticating Requests: Using the Authorization Header \(AWS Signature Version 4\) \(p. 16\)](#)
- [Authenticating Requests: Using Query Parameters \(AWS Signature Version 4\) \(p. 36\)](#)
- [Examples: Signature Calculations in AWS Signature Version 4 \(p. 41\)](#)
- [Authenticating Requests: Browser-Based Uploads Using POST \(AWS Signature Version 4\) \(p. 43\)](#)
- [Amazon S3 Signature Version 4 Authentication Specific Policy Keys \(p. 45\)](#)

Every interaction with Amazon S3 is either authenticated or anonymous. This section explains request authentication with the AWS Signature Version 4 algorithm.

### Note

If you use the AWS SDKs (see [Sample Code and Libraries](#)) to send your requests, you don't need to read this section because the SDK clients authenticate your requests by using access keys that you provide. Unless you have a good reason not to, you should always use the AWS SDKs. In regions that support both signature versions, you can request AWS SDKs to use specific signature version. For more information, see [Specifying Signature Version in Request Authentication](#) in the *Amazon Simple Storage Service Developer Guide*. You need to read this section only if you are implementing the AWS Signature Version 4 algorithm in your custom client.

Authentication with AWS Signature version 4 provides some or all of the following, depending on how you choose to sign your request:

- **Verification of the identity of the requester** – Authenticated requests require a signature that you create by using your access keys (access key ID, secret access key). For information about getting access keys, see [Understanding and Getting Your Security Credentials](#) in the *AWS General Reference*. If you are using temporary security credentials, the signature calculations also require a security token. For more information, see [Requesting Temporary Security Credentials](#) in the *IAM User Guide*.
- **In-transit data protection** – In order to prevent tampering with a request while it is in transit, you use some of the request elements to calculate the request signature. Upon receiving the request, Amazon S3 calculates the signature by using the same request elements. If any request component received by Amazon S3 does not match the component that was used to calculate the signature, Amazon S3 will reject the request.
- **Protect against reuse of the signed portions of the request** – The signed portions (using AWS Signatures) of requests are valid within 15 minutes of the timestamp in the request. An unauthorized party who has access to a signed request can modify the unsigned portions of the request without affecting the request's validity in the 15 minute window. Because of this, we recommend that you maximize protection by signing request headers and body, making HTTPS requests to Amazon S3, and by using the `s3:x-amz-content-sha256` condition key (see [Amazon S3 Signature Version 4 Authentication Specific Policy Keys \(p. 45\)](#)) in AWS policies to require users to sign S3 request bodies.

### Note

Amazon S3 supports Signature Version 4, a protocol for authenticating inbound API requests to AWS services, in all AWS regions. At this time, AWS regions created before January 30, 2014 will

continue to support the previous protocol, Signature Version 2. Any new regions after January 30, 2014 will support only Signature Version 4 and therefore all requests to those regions must be made with Signature Version 4. For more information about AWS Signature Version 2, see [Signing and Authenticating REST Requests](#) in the *Amazon Simple Storage Service Developer Guide*.

## Authentication Methods

You can express authentication information by using one of the following methods:

- **HTTP Authorization header** – Using the HTTP Authorization header is the most common method of authenticating an Amazon S3 request. All of the Amazon S3 REST operations (except for browser-based uploads using POST requests) require this header. For more information about the Authorization header value, and how to calculate signature and related options, see [Authenticating Requests: Using the Authorization Header \(AWS Signature Version 4\) \(p. 16\)](#).
- **Query string parameters** – You can use a query string to express a request entirely in a URL. In this case, you use query parameters to provide request information, including the authentication information. Because the request signature is part of the URL, this type of URL is often referred to as a presigned URL. You can use presigned URLs to embed clickable links, which can be valid for up to seven days, in HTML. For more information, see [Authenticating Requests: Using Query Parameters \(AWS Signature Version 4\) \(p. 36\)](#).

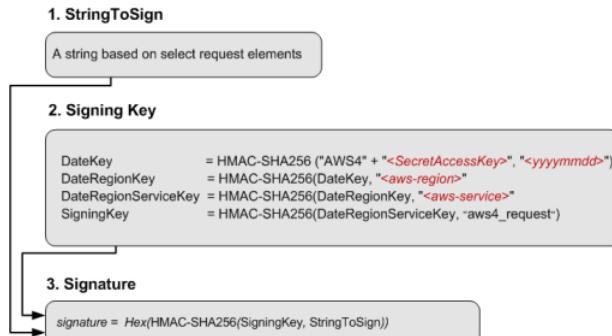
Amazon S3 also supports browser-based uploads that use an HTTP POST requests. With an HTTP POST request, you can upload content to Amazon S3 directly from the browser. For information about authenticating POST requests, see [Browser-Based Uploads Using POST](#) in the *Amazon Simple Storage Service Developer Guide*.

## Introduction to Signing Requests

Authentication information that you send in a request must include a signature. To calculate a signature, you first concatenate select request elements to form a string, referred to as the *string to sign*. You then use a signing key to calculate the hash-based message authentication code (HMAC) of the string to sign.

In AWS Signature Version 4, you don't use your secret access key to sign the request. Instead, you first use your secret access key to create a signing key. The signing key is scoped to a specific region and service, and it never expires.

The following diagram illustrates the general process of computing a signature.



The string to sign depends on the request type. For example, when you use the HTTP Authorization header or the query parameters for authentication, you use a varying combination of request elements to create the string to sign. For an HTTP POST request, the POST policy in the request is the string you

sign. For more information about computing string to sign, follow links provided at the end of this section.

For signing key, the diagram shows series of calculations, where result of each step you feed into the next step. The final step is the signing key.

Upon receiving an authenticated request, Amazon S3 servers re-create the signature by using the authentication information that is contained in the request. If the signatures match, Amazon S3 processes your request; otherwise, the request is rejected.

For more information about authenticating requests, see the following topics:

- [Authenticating Requests: Using the Authorization Header \(AWS Signature Version 4\) \(p. 16\)](#)
- [Authenticating Requests: Using Query Parameters \(AWS Signature Version 4\) \(p. 36\)](#)
- [Authenticating Requests in Browser-Based Uploads Using POST \(AWS Signature Version 4\) \(p. 49\)](#)

## Authenticating Requests: Using the Authorization Header (AWS Signature Version 4)

### Topics

- [Overview \(p. 16\)](#)
- [Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk \(AWS Signature Version 4\) \(p. 18\)](#)
- [Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks \(Chunked Upload\) \(AWS Signature Version 4\) \(p. 29\)](#)

## Overview

Using the HTTP Authorization header is the most common method of providing authentication information. Except for [POST requests \(p. 385\)](#) and requests that are signed by using query parameters, all Amazon S3 [bucket operations \(p. 76\)](#) and [object operations \(p. 333\)](#) use the Authorization request header to provide authentication information.

The following is an example of the Authorization header value. Line breaks are added to this example for readability:

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/aws4_request,
SignedHeaders=host;range;x-amz-date,
Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024
```

The following table describes the various components of the Authorization header value in the preceding example:

Component	Description
AWS4-HMAC-SHA256	The algorithm that was used to calculate the signature. You must provide this value when you use AWS Signature Version 4 for authentication.  The string specifies AWS Signature Version 4 (AWS4) and the signing algorithm (HMAC-SHA256).

Component	Description
Credential	<p>Your access key ID and the scope information, which includes the date, region, and service that were used to calculate the signature.</p> <p>This string has the following form:</p> <pre style="border: 1px solid black; padding: 5px;"><code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</code></pre> <p>Where:</p> <ul style="list-style-type: none"> <li>• <code>&lt;date&gt;</code> value is specified using YYYYMMDD format.</li> <li>• <code>&lt;aws-service&gt;</code> value is s3 when sending request to Amazon S3.</li> </ul>
SignedHeaders	<p>A semicolon-separated list of request headers that you used to compute Signature. The list includes header names only, and the header names must be in lowercase. For example:</p> <pre style="border: 1px solid black; padding: 5px;"><code>host;range;x-amz-date</code></pre>
Signature	<p>The 256-bit signature expressed as 64 lowercase hexadecimal characters. For example:</p> <pre style="border: 1px solid black; padding: 5px;"><code>fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024</code></pre> <p>Note that the signature calculations vary depending on the option you choose to transfer the payload.</p>

The signature calculations vary depending on the method you choose to transfer the request payload. S3 supports the following options:

- **Transfer payload in a single chunk** – In this case, you have the following signature calculation options:
  - **Signed payload option** – You can optionally compute the entire payload checksum and include it in signature calculation. This provides added security but you need to read your payload twice or buffer it in memory.

For example, in order to upload a file, you need to read the file first to compute a payload hash for signature calculation and again for transmission when you create the request. For smaller payloads, this approach might be preferable. However, for large files, reading the file twice can be inefficient, so you might want to upload data in chunks instead.

We recommend you include payload checksum for added security.

- **Unsigned payload option** – Do not include payload checksum in signature calculation.

For step-by-step instructions to calculate signature and construct the Authorization header value, see [Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk \(AWS Signature Version 4\) \(p. 18\)](#).

- **Transfer payload in multiple chunks (chunked upload)** – In this case you transfer payload in chunks. You can transfer a payload in chunks regardless of the payload size.

You can break up your payload into chunks. These can be fixed or variable-size chunks. By uploading data in chunks, you avoid reading the entire payload to calculate the signature. Instead, for the first chunk, you calculate a seed signature that uses only the request headers. The second chunk contains

the signature for the first chunk, and each subsequent chunk contains the signature for the chunk that precedes it. At the end of the upload, you send a final chunk with 0 bytes of data that contains the signature of the last chunk of the payload. For more information, see [Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks \(Chunked Upload\) \(AWS Signature Version 4\) \(p. 29\)](#).

When you send a request, you must tell Amazon S3 which of the preceding options you have chosen in your signature calculation, by adding the `x-amz-content-sha256` header with one of the following values:

- If you choose chunked upload options, set the header value to `STREAMING-AWS4-HMAC-SHA256-PAYLOAD`.
- If you choose to upload payload in a single chunk, set the header value to the payload checksum (`signed payload` option), or set the value to the literal string `UNSIGNED-PAYLOAD` (`unsigned payload` option).

Upon receiving the request, Amazon S3 re-creates the string to sign using information in the `Authorization` header and the `Date` header. It then verifies with authentication service the signatures match. The request date can be specified by using either the HTTP `Date` or the `x-amz-date` header. If both headers are present, `x-amz-date` takes precedence.

If the signatures match, Amazon S3 processes your request; otherwise, your request will fail.

For more information, see the following topics:

[Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk \(AWS Signature Version 4\) \(p. 18\)](#)

[Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks \(Chunked Upload\) \(AWS Signature Version 4\) \(p. 29\)](#)

## Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4)

When using the `Authorization` header to authenticate requests, the header value includes, among other things, a signature. The signature calculations vary depending on the choice you make for transferring the payload ([Overview \(p. 16\)](#)). This section explains signature calculations when you choose to transfer the payload in a single chunk. The example section (see [Examples: Signature Calculations \(p. 23\)](#)) shows signature calculations and resulting `Authorization` headers that you can use as a test suite to verify your code.

### Important

When transferring payload in a single chunk, you can optionally choose to include the payload hash in the signature calculations, referred as *signed payload* (if you don't include it, the payload is considered *unsigned*). The signing procedure discussed in the following section applies to both, but note the following differences:

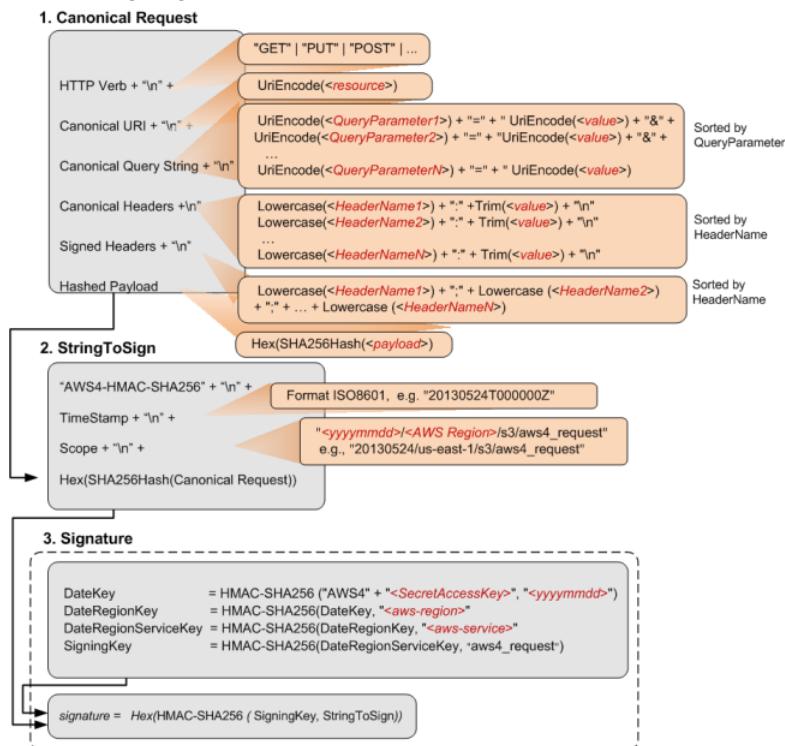
- **Signed payload option** – You include the payload hash when constructing the canonical request (that then becomes part of `StringToSign`, as explained in the signature calculation section). You also specify the same value as the `x-amz-content-sha256` header value when sending the request to S3.
- **Unsigned payload option** – You include the literal string `UNSIGNED-PAYLOAD` when constructing a canonical request, and set the same value as the `x-amz-content-sha256` header value when sending the request to S3.

When you send your request to S3, the `x-amz-content-sha256` header value informs S3 whether the payload is signed or not. Amazon S3 can then create signature accordingly for verification.

## Calculating a Signature

To calculate a signature, you first need a string to sign. You then calculate a `HMAC-SHA256` hash of the string to sign by using a signing key. The following diagram illustrates the process, including the various components of the string that you create for signing

When Amazon S3 receives an authenticated request, it computes the signature and then compares it with the signature that you provided in the request. For that reason, you must compute the signature by using the same method that is used by Amazon S3. The process of putting a request in an agreed-upon form for signing is called canonicalization.



The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

Function	Description
<code>Lowercase()</code>	Convert the string to lowercase.
<code>Hex()</code>	Lowercase base 16 encoding.
<code>SHA256Hash()</code>	Secure Hash Algorithm (SHA) cryptographic hash function.
<code>HMAC-SHA256()</code>	Computes HMAC by using the SHA256 algorithm with the signing key provided. This is the final signature.
<code>Trim()</code>	Remove any leading or trailing whitespace.
<code>UriEncode()</code>	URI encode every byte. <code>UriEncode()</code> must enforce the following rules:

Function	Description
	<ul style="list-style-type: none"> <li>• URI encode every byte except the unreserved characters: 'A'-'Z', 'a'-'z', '0'-'9', '!', '.', '_', and '~'.</li> <li>• The space character is a reserved character and must be encoded as "%20" (and not as "+").</li> <li>• Each URI encoded byte is formed by a '%' and the two-digit hexadecimal value of the byte.</li> <li>• Letters in the hexadecimal value must be uppercase, for example "%1A".</li> <li>• Encode the forward slash character, '/', everywhere except in the object key name. For example, if the object key name is photos/Jan/sample.jpg, the forward slash in the key name is not encoded.</li> </ul> <p><b>Important</b> The standard UriEncode functions provided by your development platform may not work because of differences in implementation and related ambiguity in the underlying RFCs. We recommend that you write your own custom UriEncode function to ensure that your encoding will work.</p> <p>The following is an example UriEncode() function in Java.</p> <pre>public static String UriEncode(CharSequence input, boolean encodeSlash) {     StringBuilder result = new StringBuilder();     for (int i = 0; i &lt; input.length(); i++) {         char ch = input.charAt(i);         if ((ch &gt;= 'A' &amp;&amp; ch &lt;= 'Z')    (ch &gt;= 'a' &amp;&amp; ch &lt;= 'z')    (ch &gt;= '0' &amp;&amp; ch &lt;= '9')    ch == '_'    ch == '-'    ch == '~'    ch == '.') {             result.append(ch);         } else if (ch == '/') {             result.append(encodeSlash ? "%2F" : ch);         } else {             result.append(toHexUTF8(ch));         }     }     return result.toString(); }</pre>

## Task 1: Create a Canonical Request

This section provides an overview of creating a canonical request.

The following is the canonical request format that Amazon S3 uses to calculate a signature. For signatures to match, you must create a canonical request in this format:

```
<HTTPMethod>\n<CanonicalURI>\n<CanonicalQueryString>\n<CanonicalHeaders>\n<SignedHeaders>\n<HashedPayload>
```

Where:

- **HTTPMethod** is one of the HTTP methods, for example GET, PUT, HEAD, and DELETE.
- **CanonicalURI** is the URI-encoded version of the absolute path component of the URI—everything starting with the "/" that follows the domain name and up to the end of the string or to the question mark character (?) if you have query string parameters. The URI in the following example, /examplebucket/myphoto.jpg, is the absolute path and you don't encode the "/" in the absolute path:

```
http://s3.amazonaws.com/examplebucket/myphoto.jpg
```

#### Note

You do not normalize URI paths for requests to Amazon S3. For example, you may have a bucket with an object named "my-object//example//photo.user". Normalizing the path changes the object name in the request to "my-object/example/photo.user". This is an incorrect path for that object.

- **CanonicalQueryString** specifies the URI-encoded query string parameters. You URI-encode name and values individually. You must also sort the parameters in the canonical query string alphabetically by key name. The sorting occurs after encoding. The query string in the following URI example is prefix=somePrefix&marker=someMarker&max-keys=20:

```
http://s3.amazonaws.com/examplebucket?prefix=somePrefix&marker=someMarker&max-keys=20
```

The canonical query string is as follows (line breaks are added to this example for readability):

```
UriEncode("marker")+"="+UriEncode("someMarker")+"&" +  
UriEncode("max-keys")+"="+UriEncode("20") + "&" +  
UriEncode("prefix")+"="+UriEncode("somePrefix")
```

When a request targets a subresource, the corresponding query parameter value will be an empty string (""). For example, the following URI identifies the **ACL** subresource on the **examplebucket** bucket:

```
http://s3.amazonaws.com/examplebucket?acl
```

The CanonicalQueryString in this case is as follows:

```
UriEncode("acl") + "=" + ""
```

If the URI does not include a '?', there is no query string in the request, and you set the canonical query string to an empty string (""). You will still need to include the "\n".

- **CanonicalHeaders** is a list of request headers with their values. Individual header name and value pairs are separated by the newline character ("\n"). Header names must be in lowercase. You must sort the header names alphabetically to construct the string, as shown in the following example:

```
Lowercase(<HeaderName1>)+":"+Trim(<value>)+"\n"  
Lowercase(<HeaderName2>)+":"+Trim(<value>)+"\n"  
...  
Lowercase(<HeaderNameN>)+":"+Trim(<value>)+"\n"
```

The **Lowercase()** and **Trim()** functions used in this example are described in the preceding section.

The **CanonicalHeaders** list must include the following:

---

API Version 2006-03-01

- HTTP host header.
- If the Content-Type header is present in the request, you must add it to the *CanonicalHeaders* list.
- Any x-amz-\* headers that you plan to include in your request must also be added. For example, if you are using temporary security credentials, you need to include x-amz-security-token in your request. You must add this header in the list of *CanonicalHeaders*.

**Note**

The x-amz-content-sha256 header is required for all AWS Signature Version 4 requests. It provides a hash of the request payload. If there is no payload, you must provide the hash of an empty string.

The following is an example CanonicalHeaders string. The header names are in lowercase and sorted.

```
host:s3.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b785
2b855
x-amz-date:20130708T220855Z
```

**Note**

For the purpose of calculating an authorization signature, only the host and any x-amz-\* headers are required; however, in order to prevent data tampering, you should consider including all the headers in the signature calculation.

- *SignedHeaders* is an alphabetically sorted, semicolon-separated list of lowercase request header names. The request headers in the list are the same headers that you included in the CanonicalHeaders string. For example, for the previous example, the value of *SignedHeaders* would be as follows:

```
host;x-amz-content-sha256;x-amz-date
```

- *HashedPayload* is the hexadecimal value of the SHA256 hash of the request payload.

```
Hex(SHA256Hash(<payload>))
```

If there is no payload in the request, you compute a hash of the empty string as follows:

```
Hex(SHA256Hash(""))
```

The hash returns the following value:

```
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

For example, when you upload an object by using a PUT request, you provide object data in the body. When you retrieve an object by using a GET request, you compute the empty string hash.

## Task 2: Create a String to Sign

This section provides an overview of creating a string to sign. For step-by-step instructions, see [Task 2: Create a String to Sign](#) in the *AWS General Reference*.

The string to sign is a concatenation of the following strings:

```
"AWS4-HMAC-SHA256" + "\n" +
```

```
timeStampISO8601Format + "\n" +
<Scope> + "\n" +
Hex(SHA256Hash(<CanonicalRequest>))
```

The constant string AWS4-HMAC-SHA256 specifies the hash algorithm that you are using, HMAC-SHA256. The timeStamp is the current UTC time in ISO 8601 format (for example, 20130524T000000Z).

Scope binds the resulting signature to a specific date, an AWS region, and a service. Thus, your resulting signature will work only in the specific region and for a specific service. The signature is valid for seven days after the specified date.

```
date.Format(<YYYYMMDD>) + "/" + <region> + "/" + <service> + "/aws4_request"
```

For Amazon S3, the service string is s3. For a list of *region* strings, see [Regions and Endpoints](#) in the *AWS General Reference*. The region column in this table provides the list of valid region strings.

The following scope restricts the resulting signature to the us-east-1 region and Amazon S3.

```
20130606/us-east-1/s3/aws4_request
```

#### Note

Scope must use the same date that you use to compute the signing key, as discussed in the following section.

### Task 3: Calculate Signature

In AWS Signature Version 4, instead of using your AWS access keys to sign a request, you first create a signing key that is scoped to a specific region and service. For more information about signing keys, see [Introduction to Signing Requests \(p. 15\)](#).

```
DateKey           = HMAC-SHA256("AWS4" + <SecretAccessKey>, "<YYYYMMDD>")
DateRegionKey     = HMAC-SHA256(<DateKey>, "<aws-region>")
DateRegionServiceKey = HMAC-SHA256(<DateRegionKey>, "<aws-service>")
SigningKey        = HMAC-SHA256(<DateRegionServiceKey>, "aws4_request")
```

#### Note

This signing key is valid for seven days from the date specified in the DateKey hash.

For a list of region strings, see [Regions and Endpoints](#) in the *AWS General Reference*.

Using a signing key enables you to keep your AWS credentials in one safe place. For example, if you have multiple servers that communicate with Amazon S3, you share the signing key with those servers; you don't have to keep a copy of your secret access key on each server. Signing key is valid for up to seven days. So each time you calculate signing key you will need to share the signing key with your servers. For more information, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

The final signature is the HMAC-SHA256 hash of the string to sign, using the signing key as the key.

```
HMAC-SHA256(SigningKey, StringToSign)
```

For step-by-step instructions on creating a signature, see [Task 3: Create a Signature](#) in the *AWS General Reference*.

### Examples: Signature Calculations

You can use the examples in this section as a reference to check signature calculations in your code. For additional references, see [Signature Version 4 Test Suite](#) of the *AWS General Reference*. The calculations shown in the examples use the following data:

- Example access keys.

Parameter	Value
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE
AWSSecretAccessKey	wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

- Request timestamp of 20130524T000000Z (*Fri, 24 May 2013 00:00:00 GMT*).
- Bucket name `examplebucket`.
- The bucket is assumed to be in the US East (N. Virginia) region. The credential Scope and the Signing Key calculations use `us-east-1` as the region specifier. For information about other regions, see [Regions and Endpoints](#) in the *AWS General Reference*.
- You can use either path-style or virtual hosted-style requests. The following examples show how to sign a virtual hosted-style request, for example:

```
https://examplebucket.s3.amazonaws.com/photos/photo1.jpg
```

For more information, see [Virtual Hosting of Buckets](#) in the *Amazon Simple Storage Service Developer Guide*.

### Example: GET Object

The following example gets the first 10 bytes of an object (`test.txt`) from `examplebucket`. For more information about the API action, see [GET Object \(p. 349\)](#).

```
GET /test.txt HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date:20130524T000000Z
Authorization: SignatureToBeCalculated
Range: bytes=0-9
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date: 20130524T000000Z
```

Because this GET request does not provide any body content, the `x-amz-content-sha256` value is the hash of the empty request body. The following steps show signature calculations and construction of the `Authorization` header.

#### 1. StringToSign

##### a. CanonicalRequest

```
GET
/test.txt

host:examplebucket.s3.amazonaws.com
range:bytes=0-9
x-amz-content-
sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130524T000000Z

host;range;x-amz-content-sha256;x-amz-date
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

In the canonical request string, the last line is the hash of the empty request body. The third line is empty because there are no query parameters in the request.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
7344ae5b7ee6c3e7e6b0fe0640412a37625d1fbfff95c48bbb2dc43964946972
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>","20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```
f0e8bdb87c964420e857bd35b5d6ed310bd44f0170aba48dd91039c6036bdb41
```

4. **Authorization header**

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/
s3/aws4_request, SignedHeaders=host;range;x-amz-content-sha256;x-amz-
date, Signature=f0e8bdb87c964420e857bd35b5d6ed310bd44f0170aba48dd91039c6036bdb41
```

**Example: PUT Object**

This example PUT request creates an object (`test$file.text`) in `examplebucket`. The example assumes the following:

- You are requesting `REDUCED_REDUNDANCY` as the storage class by adding the `x-amz-storage-class` request header. For information about storage classes, see [Storage Classes](#) in the *Amazon Simple Storage Service Developer Guide*.
- The content of the uploaded file is a string, "Welcome to Amazon S3." The value of `x-amz-content-sha256` in the request is based on this string.

For information about the API action, see [PUT Object \(p. 412\)](#).

```
PUT test$file.text HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Fri, 24 May 2013 00:00:00 GMT
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-storage-class: REDUCED_REDUNDANCY
x-amz-content-sha256: 44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4b4e99e42034a8b803f8b072
<Payload>
```

The following steps show signature calculations.

1. **StringToSign**

a. **CanonicalRequest**

```
PUT
/test%24file.text

date:Fri, 24 May 2013 00:00:00 GMT
```

```
host:examplebucket.s3.amazonaws.com
x-amz-content-sha256:44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4b4e99e42034a8b803f8b072
x-amz-date:20130524T000000Z
x-amz-storage-class:REDUCED_REDUNDANCY

date;host;x-amz-content-sha256;x-amz-date;x-amz-storage-class
44ce7dd67c959e0d3524ffac1771dfbba87d2b6b4b4e99e42034a8b803f8b072
```

In the canonical request, the third line is empty because there are no query parameters in the request. The last line is the hash of the body, which should be same as the `x-amz-content-sha256` header value.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
9e0e90d9c76de8fa5b200d8c849cd5b8dc7a3be3951ddb7f6a76b4158342019d
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>","20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```
98ad721746da40c64f1a55b78f14c238d841ea1380cd77a1b5971af0ece108bd
```

4. **Authorization header**

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/
aws4_request,SignedHeaders=date;host;x-amz-content-sha256;x-amz-date;x-amz-storage-
class,Signature=98ad721746da40c64f1a55b78f14c238d841ea1380cd77a1b5971af0ece108bd
```

## Example: GET Bucket Lifecycle

The following GET request retrieves the lifecycle configuration of `examplebucket`. For information about the API action, see [GET Bucket lifecycle \(p. 145\)](#).

```
GET ?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Because the request does not provide any body content, the `x-amz-content-sha256` header value is the hash of the empty request body. The following steps show signature calculations.

1. **StringToSign**

a. **CanonicalRequest**

```
GET
/
lifecycle=
host:examplebucket.s3.amazonaws.com
```

```
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130524T000000Z

host;x-amz-content-sha256;x-amz-date
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

In the canonical request, the last line is the hash of the empty request body.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
9766c798316ff2757b517bc739a67f6213b4ab36dd5da2f94eaebf79c77395ca
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>","20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```
fea454ca298b7da1c68078a5d1bdbfbbe0d65c699e0f91ac7a200a0136783543
```

4. **Authorization header**

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/
s3/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-
date, Signature=fea454ca298b7da1c68078a5d1bdbfbbe0d65c699e0f91ac7a200a0136783543
```

## Example: Get Bucket (List Objects)

The following example retrieves a list of objects from examplebucket bucket. For information about the API action, see [GET Bucket \(List Objects\) Version 1 \(p. 111\)](#).

```
GET ?max-keys=2&prefix=J HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Authorization: SignatureToBeCalculated
x-amz-date: 20130524T000000Z
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Because the request does not provide a body, the value of x-amz-content-sha256 is the hash of the empty request body. The following steps show signature calculations.

1. **StringToSign**

a. **CanonicalRequest**

```
GET
/
max-keys=2&prefix=J
host:examplebucket.s3.amazonaws.com
x-amz-content-
sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20130524T000000Z
```

```
host;x-amz-content-sha256;x-amz-date
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

In the canonical string, the last line is the hash of the empty request body.

b. **StringToSign**

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
df57d21db20da04d7fa30298dd4488ba3a2b47ca3a489c74750e0f1e7df1b9b7
```

2. **SigningKey**

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>","20130524"),"us-east-1"),"s3"),"aws4_request")
```

3. **Signature**

```
34b48302e7b5fa45bde8084f4b7868a86f0a534bc59db6670ed5711ef69dc6f7
```

4. **Authorization header**

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/
s3/aws4_request,SignedHeaders=host;x-amz-content-sha256;x-amz-
date,Signature=34b48302e7b5fa45bde8084f4b7868a86f0a534bc59db6670ed5711ef69dc6f7
```

# Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4)

As described in the [Overview \(p. 16\)](#), when authenticating requests using the `Authorization` header, you have an option of uploading the payload in chunks. You can send data in fixed size or variable size chunks. This section describes the signature calculation process in chunked upload, how you create the chunk body, and how the delayed signing works where you first upload the chunk, and send its signature in the subsequent chunk. The example section (see [Example: PUT Object \(p. 33\)](#)) shows signature calculations and resulting `Authorization` headers that you can use as a test suite to verify your code.

## Note

When transferring data in a series of chunks, you must use the `Content-Length` HTTP header to explicitly specify the total content length (object length in bytes plus metadata in each chunk). This requires you to pre-compute the total length of the payload, including the metadata you send in each chunk, before starting your request. The `x-amz-decoded-content-length` header contains the size of the object length in bytes.

Each chunk signature calculation includes the signature of the previous chunk. To begin, you create a *seed* signature using only the headers. You use the seed signature in the signature calculation of the first chunk. For each subsequent chunk, you create a chunk signature that includes the signature of the previous chunk. Thus, the chunk signatures are chained together; that is, the signature of chunk  $n$  is a function  $F(chunk\ n, signature(chunk\ n-1))$ . The chaining ensures that you send the chunks in the correct order.

To perform a chunked upload, do the following:

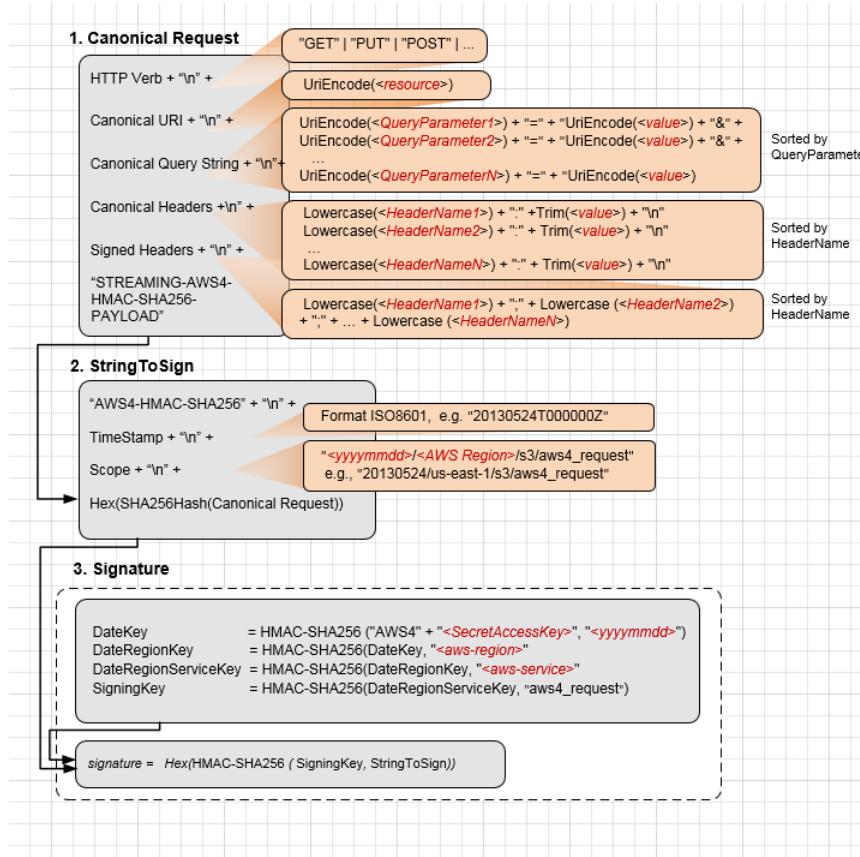
1. Decide the payload chunk size. You need this when you write the code.

Chunk size must be at least 8 KB. We recommend a chunk size of at least 64 KB for better performance. This chunk size applies to all chunks except the last one. The last chunk you send can be smaller than 8 KB. If your payload is small and can fit into one chunk, then it can be smaller than the 8 KB.

2. Create the seed signature for inclusion in the first chunk. For more information, see [Calculating the Seed Signature \(p. 29\)](#).
3. Create the first chunk and stream it. For more information, see [Defining the Chunk Body \(p. 32\)](#).
4. For each subsequent chunk, calculate the chunk signature that includes the previous signature in the string you sign, construct the chunk, and send it. For more information, see [Defining the Chunk Body \(p. 32\)](#).
5. Send the final additional chunk, which is the same as the other chunks in the construction, but it has zero data bytes. For more information, see [Defining the Chunk Body \(p. 32\)](#).

## Calculating the Seed Signature

The following diagram illustrates the process of calculating the seed signature.



The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

Function	Description
Lowercase()	Convert the string to lowercase.
Hex()	Lowercase base 16 encoding.
SHA256Hash()	Secure Hash Algorithm (SHA) cryptographic hash function.
HMAC-SHA256()	Computes HMAC by using the SHA256 algorithm with the signing key provided. This is the final signature.
Trim()	Remove any leading or trailing whitespace.
UriEncode()	URI encode every byte. UriEncode() must enforce the following rules: <ul style="list-style-type: none"> <li>URI encode every byte except the unreserved characters: 'A'-'Z', 'a'-'z', '0'-'9', '-', '.', '_', and '~'.</li> <li>The space character is a reserved character and must be encoded as "%20" (and not as "+").</li> <li>Each URI encoded byte is formed by a '%' and the two-digit hexadecimal value of the byte.</li> <li>Letters in the hexadecimal value must be uppercase, for example "%1A".</li> </ul>

Function	Description
	<ul style="list-style-type: none"> <li>Encode the forward slash character, '/', everywhere except in the object key name. For example, if the object key name is photos/Jan/sample.jpg, the forward slash in the key name is not encoded.</li> </ul> <p><b>Important</b> The standard UriEncode functions provided by your development platform may not work because of differences in implementation and related ambiguity in the underlying RFCs. We recommend that you write your own custom UriEncode function to ensure that your encoding will work.</p> <p>The following is an example UriEncode() function in Java.</p> <pre>public static String UriEncode(CharSequence input, boolean encodeSlash) {     StringBuilder result = new StringBuilder();     for (int i = 0; i &lt; input.length(); i++) {         char ch = input.charAt(i);         if ((ch &gt;= 'A' &amp;&amp; ch &lt;= 'Z')    (ch &gt;= 'a' &amp;&amp; ch &lt;= 'z')    (ch &gt;= '0' &amp;&amp; ch &lt;= '9')    ch == '_'    ch == '-'    ch == '~'    ch == '.') {             result.append(ch);         } else if (ch == '/') {             result.append(encodeSlash ? "%2F" : ch);         } else {             result.append(toHexUTF8(ch));         }     }     return result.toString(); }</pre>

For information about the signing process, see [Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk \(AWS Signature Version 4\)](#) (p. 18). The process is the same, except that the creation of CanonicalRequest differs as follows:

- In addition to the request headers you plan to add, you must include the following headers:

Header	Description
x-amz-content-sha256	This header is required for all AWS Signature Version 4 requests. Set the value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD to indicate that the signature covers only headers and that there is no payload.

Header	Description
Content-Encoding	<p>Set the value to <code>aws-chunked</code>.</p> <p>Amazon S3 supports multiple content encodings. For example:</p> <pre style="border: 1px solid black; padding: 5px;">Content-Encoding : aws-chunked,gzip</pre> <p>That is, you can specify your custom content-encoding when using Signature Version 4 streaming API.</p> <p><b>Note</b> Amazon S3 stores the resulting object without the <code>aws-chunked</code> encoding. Therefore, when you retrieve the object, it is not <code>aws-chunked</code> encoded.</p>
<code>x-amz-decoded-content-length</code>	Set the value to the length, in bytes, of the data to be chunked, without counting any metadata. For example, if you are uploading a 4 GB file, set the value to 4294967296. This is the raw size of the object to be uploaded (data you want to store in Amazon S3).
Content-Length	Set the value to the actual size of the transmitted HTTP body, which includes the length of your data (value set for <code>x-amz-decoded-content-length</code> ) plus, chunk metadata. Each chunk has metadata, such as the signature of the previous chunk. Chunk calculations are discussed in the following section.

You send the first chunk with the seed signature. You must construct the chunk as described in the following section.

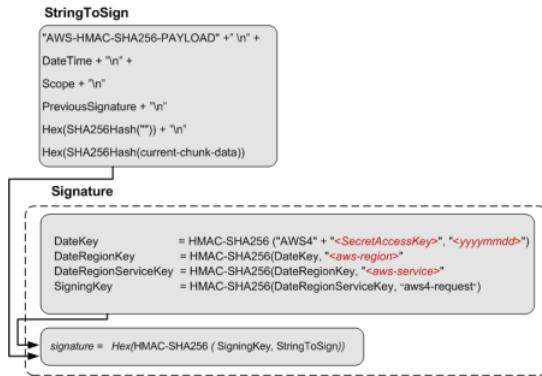
## Defining the Chunk Body

All chunks include some metadata. Each chunk must conform to the following structure:

```
string(IntHexBase(chunk-size)) + ";chunk-signature=" + signature + \r\n + chunk-data + \r\n
```

Where:

- `IntHexBase()` is a function that you write to convert an integer chunk-size to hexadecimal. For example, if chunk-size is 65536, hexadecimal string is "10000".
- `chunk-size` is the size, in bytes, of the chunk-data, without metadata. For example, if you are uploading a 65 KB object and using a chunk size of 64 KB, you upload the data in three chunks: the first would be 64 KB, the second 1 KB, and the final chunk with 0 bytes.
- `signature` For each chunk, you calculate the signature using the following string to sign. For the first chunk, you use the seed-signature as the previous signature.



The size of the final chunk data that you send is 0, although the chunk body still contains metadata, including the signature of the previous chunk.

## Example: PUT Object

You can use the examples in this section as a reference to check signature calculations in your code. Before you review the examples, note the following:

- The signature calculations in these examples use the following example security credentials.

Parameter	Value
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE
AWSSecretAccessKey	wJalrXUtnFEMI/K7MDENG/bPxRficyEXAMPLEKEY

- All examples use the request time stamp 20130524T000000Z (*Fri, 24 May 2013 00:00:00 GMT*).
- All examples use `examplebucket` as the bucket name.
- The bucket is assumed to be in the US East (N. Virginia) Region, and the credential `Scope` and the `Signing Key` calculations use `us-east-1` as the Region specifier. For more information, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.
- You can use either path style or virtual-hosted style requests. The following examples use virtual-hosted style requests, for example:

`https://examplebucket.s3.amazonaws.com/photos/photo1.jpg`

For more information, see [Virtual Hosting of Buckets](#) in the *Amazon Simple Storage Service Developer Guide*.

## Example: PUT Object

The following example sends a `PUT` request to upload an object. The signature calculations assume the following:

- You are uploading a 65 KB text file, and the file content is a one-character string made up of the letter '`'a'`'.
- The chunk size is 64 KB. As a result, the payload is uploaded in three chunks, 64 KB, 1 KB, and the final chunk with 0 bytes of chunk data.
- The resulting object has the key name `chunkObject.txt`.

- You are requesting REDUCED\_REDUNDANCY as the storage class by adding the x-amz-storage-class request header.

For information about the API action, see [PUT Object \(p. 412\)](#). The general request syntax is as follows:

```
PUT /examplebucket/chunkObject.txt HTTP/1.1
Host: s3.amazonaws.com
x-amz-date: 20130524T000000Z
x-amz-storage-class: REDUCED_REDUNDANCY
Authorization: SignatureToBeCalculated
x-amz-content-sha256: STREAMING-AWS4-HMAC-SHA256-PAYLOAD
Content-Encoding: aws-chunked
x-amz-decoded-content-length: 66560
Content-Length: 66824
<Payload>
```

The following steps show signature calculations.

## 1. Seed signature — Create String to Sign

### a. CanonicalRequest

```
PUT
/examplebucket/chunkObject.txt

content-encoding:aws-chunked
content-length:66824
host:s3.amazonaws.com
x-amz-content-sha256:STREAMING-AWS4-HMAC-SHA256-PAYLOAD
x-amz-date:20130524T000000Z
x-amz-decoded-content-length:66560
x-amz-storage-class:REDUCED_REDUNDANCY

content-encoding;content-length;host;x-amz-content-sha256;x-amz-date;x-amz-decoded-
content-length;x-amz-storage-class
STREAMING-AWS4-HMAC-SHA256-PAYLOAD
```

In the canonical request, the third line is empty because there are no query parameters in the request. The last line is the constant string provided as the value of the hashed Payload, which should be same as the value of x-amz-content-sha256 header.

### b. StringToSign

```
AWS4-HMAC-SHA256
20130524T000000Z
20130524/us-east-1/s3/aws4_request
ceef3fed04b70f867d036f722359b0b1f2f0e5dc0efadbc082b76c4c60e316455
```

### Note

For information about each of line in the string to sign, see the diagram that explains seed signature calculation.

## 2. SigningKey

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>","20130524"),"us-east-1"),"s3"),"aws4_request")
```

### 3. Seed Signature

```
4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9
```

### 4. Authorization header

The resulting Authorization header is as follows:

```
AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20130524/us-east-1/s3/  
aws4_request, SignedHeaders=content-encoding;content-length;host;x-amz-  
content-sha256;x-amz-date;x-amz-decoded-content-length;x-amz-storage-  
class, Signature=4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9
```

### 5. Chunk 1: (65536 bytes, with value 97 for letter 'a')

#### a. Chunk string to sign:

```
AWS4-HMAC-SHA256-PAYLOAD  
20130524T000000Z  
20130524/us-east-1/s3/aws4_request  
4f232c4386841ef735655705268965c44a0e4690baa4adea153f7db9fa80a0a9  
e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855  
bf718b6f653bebc184e1479f1935b8da974d701b893afcf49e701f3e2f9f9c5a
```

#### Note

For information about each line in the string to sign, see the preceding diagram that shows various components of the string to sign (for example, the last three lines are, `previous-signature`, `hash("")`, and `hash(current-chunk-data)`).

#### b. Chunk signature:

```
ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648
```

#### c. Chunk data sent:

```
10000;chunk-  
signature=ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648  
<65536-bytes>
```

### 6. Chunk 2: (1024 bytes, with value 97 for letter 'a')

#### a. Chunk string to sign:

```
AWS4-HMAC-SHA256-PAYLOAD  
20130524T000000Z  
20130524/us-east-1/s3/aws4_request  
ad80c730a21e5b8d04586a2213dd63b9a0e99e0e2307b0ade35a65485a288648  
e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855  
2edc986847e209b4016e141a6dc8716d3207350f416969382d431539bf292e4a
```

#### b. Chunk signature:

```
0055627c9e194cb4542bae2aa5492e3c1575bbb81b612b7d234b86a503ef5497
```

#### c. Chunk data sent:

```
400;chunk-  
signature=0055627c9e194cb4542bae2aa5492e3c1575bbb81b612b7d234b86a503ef5497  
<1024 bytes>
```

## 7. Chunk 3: (0 byte data)

- a. Chunk string to sign:

```
AWS4-HMAC-SHA256-PAYLOAD
20130524T000000Z
20130524/us-east-1/s3/aws4_request
0055627c9e194cb4542bae2aa5492e3c1575bbb81b612b7d234b86a503ef5497
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

- b. Chunk signature:

```
b6c6ea8a5354eaf15b3cb7646744f4275b71ea724fed81ceb9323e279d449df9
```

- c. Chunk data sent:

```
0;chunk-signature=b6c6ea8a5354eaf15b3cb7646744f4275b71ea724fed81ceb9323e279d449df9
```

# Authenticating Requests: Using Query Parameters (AWS Signature Version 4)

As described in the authentication overview (see [Authentication Methods \(p. 15\)](#)), you can provide authentication information using query string parameters. Using query parameters to authenticate requests is useful when you want to express a request entirely in a URL. This method is also referred as presigning a URL.

A use case scenario for presigned URLs is that you can grant temporary access to your Amazon S3 resources. For example, you can embed a presigned URL on your website or alternatively use it in command line client (such as Curl) to download objects.

The following is an example presigned URL.

```
https://s3.amazonaws.com/examplebucket/test.txt
?X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=<your-access-key-id>/20130721/us-east-1/s3/aws4_request
&X-Amz-Date=20130721T201207Z
&X-Amz-Expires=86400
&X-Amz-SignedHeaders=host
&X-Amz-Signature=<signature-value>
```

In the example URL, note the following:

- The line feeds are added for readability.
- The X-Amz-Credential value in the URL shows the "/" character only for readability. In practice, it should be encoded as %2F. For example:

```
&X-Amz-Credential=<your-access-key-id>%2F20130721%2Fus-east-1%2Fs3%2Faws4_request
```

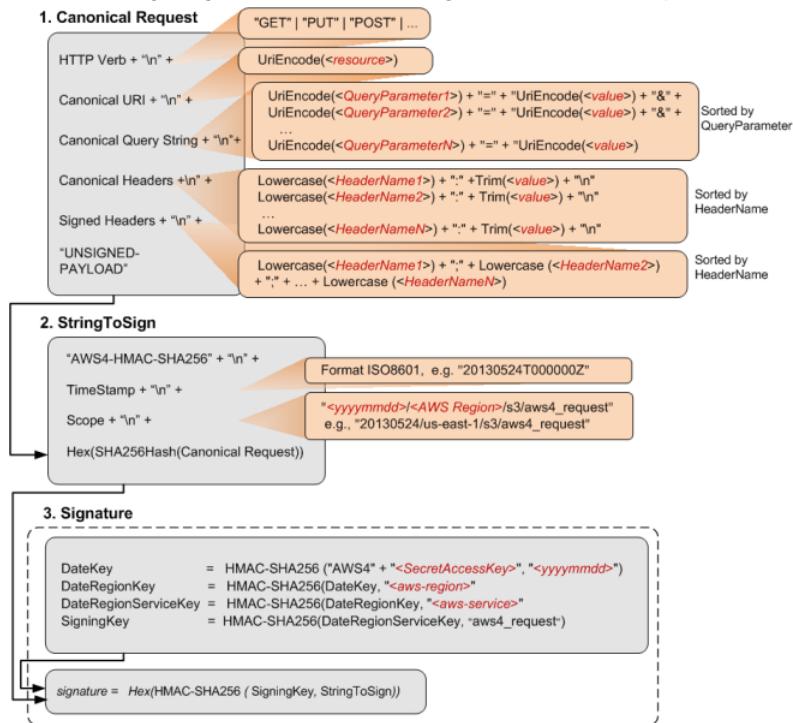
The following table describes the query parameters in the URL that provide authentication information.

Query String Parameter Name	Example Value
X-Amz-Algorithm	<p>Identifies the version of AWS Signature and the algorithm that you used to calculate the signature.</p> <p>For AWS Signature Version 4, you set this parameter value to AWS4-HMAC-SHA256. This string identifies AWS Signature Version 4 (AWS4) and the HMAC-SHA256 algorithm (HMAC-SHA256).</p>
X-Amz-Credential	<p>In addition to your access key ID, this parameter also provides scope (AWS region and service) for which the signature is valid. This value must match the scope you use in signature calculations, discussed in the following section. The general form for this parameter value is as follows:</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;AWS-region&gt;/&lt;AWS-service&gt;/aws4_request</code> </div> <p>For example:</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <code>AKIAIOSFODNN7EXAMPLE/20130721/us-east-1/s3/aws4_request</code> </div> <p>For Amazon S3, the <b>AWS-service</b> string is s3. For a list of S3 AWS-region strings, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p>
X-Amz-Date	<p>The date and time format must follow the ISO 8601 standard, and must be formatted with the "yyyyMddTHHmssZ" format. For example if the date and time was "08/01/2016 15:32:41.982-700" then it must first be converted to UTC (Coordinated Universal Time) and then submitted as "20160801T083241Z".</p>
X-Amz-Expires	<p>Provides the time period, in seconds, for which the generated presigned URL is valid. For example, 86400 (24 hours). This value is an integer. The minimum value you can set is 1, and the maximum is 604800 (seven days).</p> <p>A presigned URL can be valid for a maximum of seven days because the signing key you use in signature calculation is valid for up to seven days.</p>
X-Amz-SignedHeaders	<p>Lists the headers that you used to calculate the signature. The following headers are required in the signature calculations:</p> <ul style="list-style-type: none"> <li>• The HTTP host header.</li> <li>• Any x-amz-* headers that you plan to add to the request.</li> </ul> <p><b>Note</b> For added security, you should sign all the request headers that you plan to include in your request.</p>
X-Amz-Signature	<p>Provides the signature to authenticate your request. This signature must match the signature Amazon S3 calculates; otherwise, Amazon S3 denies the request. For example,</p> <p style="margin-left: 20px;"><code>733255ef022bec3f2a8701cd61d4b371f3f28c9f193a1f02279211d48d5193</code></p>

Query String Parameter Name	Example Value
	Signature calculations are described in the following section.

## Calculating a Signature

The following diagram illustrates the signature calculation process.



The following table describes the functions that are shown in the diagram. You need to implement code for these functions.

Function	Description
Lowercase()	Convert the string to lowercase.
Hex()	Lowercase base 16 encoding.
SHA256Hash()	Secure Hash Algorithm (SHA) cryptographic hash function.
HMAC-SHA256()	Computes HMAC by using the SHA256 algorithm with the signing key provided. This is the final signature.
Trim()	Remove any leading or trailing whitespace.
UriEncode()	URI encode every byte. UriEncode() must enforce the following rules: <ul style="list-style-type: none"> <li>URI encode every byte except the unreserved characters: 'A'-'Z', 'a'-'z', '0'-'9', '-', '_', and '-'.</li> <li>The space character is a reserved character and must be encoded as "%20" (and not as "+").</li> </ul>

Function	Description
	<ul style="list-style-type: none"> <li>Each URI encoded byte is formed by a '%' and the two-digit hexadecimal value of the byte.</li> <li>Letters in the hexadecimal value must be uppercase, for example "%1A".</li> <li>Encode the forward slash character, '/', everywhere except in the object key name. For example, if the object key name is photos/Jan/sample.jpg, the forward slash in the key name is not encoded.</li> </ul> <p><b>Important</b> The standard UriEncode functions provided by your development platform may not work because of differences in implementation and related ambiguity in the underlying RFCs. We recommend that you write your own custom UriEncode function to ensure that your encoding will work.</p> <p>The following is an example UriEncode() function in Java.</p> <pre>public static String UriEncode(CharSequence input, boolean encodeSlash) {     StringBuilder result = new StringBuilder();     for (int i = 0; i &lt; input.length(); i++) {         char ch = input.charAt(i);         if ((ch &gt;= 'A' &amp;&amp; ch &lt;= 'Z')    (ch &gt;= 'a' &amp;&amp; ch &lt;= 'z')    (ch &gt;= '0' &amp;&amp; ch &lt;= '9')    ch == '_'    ch == '-'    ch == '~'    ch == '.') {             result.append(ch);         } else if (ch == '/') {             result.append(encodeSlash ? "%2F" : ch);         } else {             result.append(toHexUTF8(ch));         }     }     return result.toString(); }</pre>

For more information about the signing process (details of creating a canonical request, string to sign, and signature calculations), see [Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk \(AWS Signature Version 4\) \(p. 18\)](#). The process is generally the same except that the creation of **CanonicalRequest** in a presigned URL differs as follows:

- You don't include a payload hash in the **Canonical Request**, because when you create a presigned URL, you don't know the payload content because the URL is used to upload an arbitrary payload. Instead, you use a constant string **UNSIGNED-PAYLOAD**.
- The **Canonical Query String** must include all the query parameters from the preceding table except for **X-Amz-Signature**.
- Canonical Headers** must include the HTTP host header. If you plan to include any of the **x-amz-\*** headers, these headers must also be added for signature calculation. You can optionally add all other headers that you plan to include in your request. For added security, you should sign as many headers as possible.

## An Example

Suppose you have an object `test.txt` in your `examplebucket` bucket. You want to share this object with others for a period of 24 hours (86400 seconds) by creating a presigned URL.

```
https://s3.amazonaws.com/examplebucket/test.txt
?X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20130524%2Fus-east-1%2Fs3%2Faws4_request
&X-Amz-Date=20130524T00000Z&X-Amz-Expires=86400&X-Amz-SignedHeaders=host
&X-Amz-Signature=<signature-value>
```

The following steps illustrate first the signature calculations and then construction of the presigned URL. The example makes the following additional assumptions:

- Request timestamp is `Fri, 24 May 2013 00:00:00 GMT`.
- The bucket is in the US East (N. Virginia) region, and the credential Scope and the Signing Key calculations use `us-east-1` as the region specifier. For more information, see [Regions and Endpoints](#) in the *AWS General Reference*.

You can use this example as a test case to verify the signature that your code calculates; however, you must use the same bucket name, object key, time stamp, and the following example credentials:

Parameter	Value
<code>AWSAccessKeyId</code>	<code>AKIAIOSFODNN7EXAMPLE</code>
<code>AWSSecretAccessKey</code>	<code>wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY</code>

### 1. StringToSign

#### a. CanonicalRequest

```
GET
/test.txt
X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE
%2F20130524%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20130524T00000Z&X-Amz-
Expires=86400&X-Amz-SignedHeaders=host
host:examplebucket.s3.amazonaws.com

host
UNSIGNED-PAYLOAD
```

#### b. StringToSign

```
AWS4-HMAC-SHA256
20130524T00000Z
20130524/us-east-1/s3/aws4_request
3bfa292879f6447bbcd7001decf97f4a54dc650c8942174ae0a9121cf58ad04
```

### 2. SigningKey

```
signing key = HMAC-SHA256(HMAC-SHA256(HMAC-SHA256(HMAC-SHA256("AWS4" +
"<YourSecretAccessKey>","20130524"), "us-east-1"), "s3"), "aws4_request")
```

### 3. Signature

```
aeeed9bccd4d02ee5c0109b86d86835f995330da4c265957d157751f604d404
```

Now you have all information to construct a presigned URL. The resulting URL for this example is shown as follows (you can use this to compare your presigned URL):

```
https://examplebucket.s3.amazonaws.com/test.txt?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE%2F20130524%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20130524T00000Z&X-Amz-Expires=86400&X-Amz-SignedHeaders=host&X-Amz-Signature=aeeed9bccd4d02ee5c0109b86d86835f995330da4c265957d157751f604d404
```

## Examples: Signature Calculations in AWS Signature Version 4

### Topics

- [Signature Calculation Examples Using Java \(AWS Signature Version 4\) \(p. 41\)](#)
- [Examples of Signature Calculations Using C# \(AWS Signature Version 4\) \(p. 42\)](#)

For authenticated requests, unless you are using the AWS SDKs, you have to write code to calculate signatures that provide authentication information in your requests. Signature calculation in AWS Signature Version 4 (see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#)) can be a complex undertaking, and we recommend that you use the AWS SDKs whenever possible.

This section provides examples of signature calculations written in Java and C#. The code samples send the following requests and use the HTTP Authorization header to provide authentication information:

- **PUT object** – Separate examples illustrate both uploading the full payload at once and uploading the payload in chunks. For information about using the Authorization header for authentication, see [Authenticating Requests: Using the Authorization Header \(AWS Signature Version 4\) \(p. 16\)](#).
- **GET object** – This example generates a presigned URL to get an object. Query parameters provide the signature and other authentication information. Users can paste a presigned URL in their browser to retrieve the object, or you can use the URL to create a clickable link. For information about using query parameters for authentication, see [Authenticating Requests: Using Query Parameters \(AWS Signature Version 4\) \(p. 36\)](#).

The rest of this section describes the examples in Java and C#. The topics include instructions for downloading the samples and for executing them.

## Signature Calculation Examples Using Java (AWS Signature Version 4)

The Java sample that shows signature calculation can be downloaded at <https://s3.amazonaws.com/aws-java-sdk/samples/AWSS3SigV4JavaSamples.jar>. In `RunAllSamples.java`, the `main()` function executes sample requests to create an object, retrieve an object, and create a presigned URL for the object. The sample creates an object from the text string provided in the code:

```
PutS3ObjectSample.putS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
GetS3ObjectSample.getS3Object(bucketName, regionName, awsAccessKey, awsSecretKey);
PresignedUrlSample.getPresignedUrlToS3Object(bucketName, regionName, awsAccessKey,
awsSecretKey);
```

```
PutS3ObjectChunkedSample.putS3ObjectChunked(bucketName, regionName, awsAccessKey,  
awsSecretKey);
```

### To test the examples on a Linux-based computer

The following instructions are for the Linux operating system.

1. At a command prompt, change the directory to the directory that contains AWSS3SigV4JavaSamples.jar.
2. Extract the source files from AWSS3SigV4JavaSamples.jar.

```
jar xvf AWSS3SigV4JavaSamples.jar
```

3. In a text editor, open the file ./com/amazonaws/services/s3/samples/RunAllSamples.java. Update code with the following information:
  - The name of a bucket where the new object can be created.

#### Note

The examples use a virtual-hosted style request to access the bucket. To avoid potential errors, ensure that your bucket name conforms to the bucket naming rules as explained in [Bucket Restrictions and Limitations](#) in the *Amazon Simple Storage Service Developer Guide*.

- AWS region where the bucket resides.

If bucket is in the US East (N. Virginia) region, use us-east-1 to specify the region. For a list of other AWS regions, go to [Amazon Simple Storage Service \(S3\)](#) in the *AWS General Reference*.

4. Compile the source code and store the compiled classes into the bin/ directory.

```
javac -d bin -source 6 -verbose com
```

5. Change the directory to bin/, and then execute RunAllSamples.

```
java com.amazonaws.services.s3.sample.RunAllSamples
```

The code runs all the methods in main(). For each request, the output will show the canonical request, the string to sign, and the signature.

## Examples of Signature Calculations Using C# (AWS Signature Version 4)

The C# sample that shows signature calculation can be downloaded at [https://docs.aws.amazon.com/AmazonS3/latest/API/samples/AmazonS3SigV4\\_Samples\\_CSharp.zip](https://docs.aws.amazon.com/AmazonS3/latest/API/samples/AmazonS3SigV4_Samples_CSharp.zip). In Program.cs, the main() function executes sample requests to create an object, retrieve an object, and create a presigned URL for the object. The code for signature calculation is in the \Signers folder.

```
PutS3ObjectSample.Run(awsRegion, bucketName, "MySampleFile.txt");  
  
Console.WriteLine("\n*****");  
PutS3ObjectChunkedSample.Run(awsRegion, bucketName, "MySampleFileChunked.txt");  
  
Console.WriteLine("\n*****");  
GetS3ObjectSample.Run(awsRegion, bucketName, "MySampleFile.txt");  
  
Console.WriteLine("\n*****");  
PresignedUrlSample.Run(awsRegion, bucketName, "MySampleFile.txt");
```

**To test the examples with Microsoft Visual Studio 2010 or later**

1. Extract the .zip file.
2. Start Visual Studio, and then open the .sln file.
3. Update the App.config file with valid security credentials.
4. Update the code as follows:
  - In Program.cs, provide the bucket name and the AWS region where the bucket resides. The sample creates an object in this bucket.
5. Execute the code.
6. To verify that the object was created, copy the presigned URL that the program creates, and then paste it in a browser window.

## Authenticating Requests: Browser-Based Uploads Using POST (AWS Signature Version 4)

Amazon S3 supports HTTP POST requests so that users can upload content directly to Amazon S3. Using HTTP POST to upload content simplifies uploads and reduces upload latency where users upload data to store in Amazon S3. This section describes how you authenticate HTTP POST requests. For more information about HTTP POST requests, how to create a form, create a POST policy, and an example, see [Authenticating Requests in Browser-Based Uploads Using POST \(AWS Signature Version 4\) \(p. 49\)](#).

To authenticate an HTTP POST request you do the following:

1. The form must include the following fields to provide signature and relevant information that Amazon S3 can use to re-calculate the signature upon receiving the request:

Element Name	Description
policy	The Base64-encoded security policy that describes what is permitted in the request. For signature calculation this policy is the string you sign. Amazon S3 must get this policy so it can re-calculate the signature.
x-amz-algorithm	The signing algorithm used. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256.
x-amz-credential	In addition to your access key ID, this provides scope information you used in calculating the signing key for signature calculation.  It is a string of the following form:  <i>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</i>  For example:  <i>AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request..</i>  For Amazon S3, the <i>aws-service</i> string is <i>s3</i> . For a list of Amazon S3 <i>aws-region</i> strings, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .

Element Name	Description
x-amz-date	<p>It is the date value in ISO8601 format. For example, 20130728T00000Z.</p> <p>It is the same date you used in creating the signing key. This must also be the same value you provide in the policy (x-amz-date) that you signed.</p>
x-amz-signature	(AWS Signature Version 4) The HMAC-SHA256 hash of the security policy.

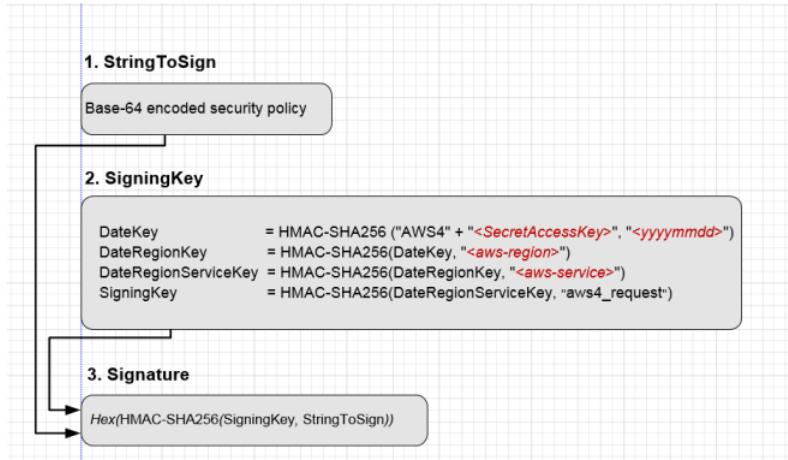
2. The POST policy must include the following elements:

Element Name	Description
x-amz-algorithm	The signing algorithm that you used to calculate the signature. For AWS Signature Version 4, the value is AWS4-HMAC-SHA256.
x-amz-credential	<p>In addition to your access key ID, this provides scope information you used in calculating the signing key for signature calculation.</p> <p>It is a string of the following form:</p> <p style="color: red;"><i>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</i></p> <p>For example,</p> <p style="color: red;">AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request..</p>
x-amz-date	The date value specified in the ISO8601 formatted string. For example, "20130728T00000Z". The date must be same that you used in creating the signing key for signature calculation.

3. For signature calculation the POST policy is the string to sign.

## Calculating a Signature

The following diagram illustrates the signature calculation process.



### To Calculate a signature

1. Create a policy using UTF-8 encoding.
2. Convert the UTF-8-encoded policy to Base64. The result is the string to sign.
3. Create the signature as an HMAC-SHA256 hash of the string to sign. You will provide the signing key as key to the hash function.
4. Encode the signature by using hex encoding.

For more information about creating HTML forms, security policies, and an example, see the following subtopics:

- [Creating an HTML Form \(Using AWS Signature Version 4\) \(p. 51\)](#)
- [Creating a POST Policy \(p. 56\)](#)
- [Example: Browser-Based Upload using HTTP POST \(Using AWS Signature Version 4\) \(p. 61\)](#)
- [Using POST with Adobe Flash to Upload Objects \(p. 63\)](#)

## Amazon S3 Signature Version 4 Authentication Specific Policy Keys

The following table shows the policy keys related Amazon S3 Signature Version 4 authentication that can be in Amazon S3 policies. In a bucket policy, you can add these conditions to enforce specific behavior when requests are authenticated by using Signature Version 4. For example policies, see [Bucket Policy Examples Using Signature Version 4 Related Condition Keys \(p. 47\)](#).

### Applicable Keys for s3:\* Actions or any of the Amazon S3 Actions

Applicable Keys	Description
s3:signatureversion	Identifies the version of AWS Signature that you want to support for authenticated requests. For authenticated requests, Amazon S3 supports both Signature Version 4 and Signature Version 2. You can add this condition in your bucket policy to require a specific signature version.  Valid values:

Applicable Keys	Description
	<p>"AWS" identifies Signature Version 2</p> <p>"AWS4-HMAC-SHA256" identifies Signature Version 4</p>
<code>s3:authType</code>	<p>Amazon S3 supports various methods of authentication (see <a href="#">Authenticating Requests (AWS Signature Version 4) (p. 14)</a>). You can optionally use this condition key to restrict incoming requests to use a specific authentication method. For example, you can allow only the HTTP Authorization header to be used in request authentication.</p> <p>Valid values:</p> <p>REST-HEADER</p> <p>REST-QUERY-STRING</p> <p>POST</p>
<code>s3:signatureAge</code>	<p>The length of time, in milliseconds, that a signature is valid in an authenticated request.</p> <p>In Signature Version 4, the signing key is valid for up to seven days (see <a href="#">Introduction to Signing Requests (p. 15)</a>). Therefore, the signatures are also valid for up to seven days. You can use this condition to further limit the signature age.</p> <p>Example value: 100</p>

Applicable Keys	Description
s3:x-amz-content-sha256	<p>You can use this condition key to disallow unsigned content in your bucket.</p> <p>When you use Signature Version 4, for requests that use the Authorization header, you add the x-amz-content-sha256 header in the signature calculation and then set its value to the hash payload.</p> <p>You can use this condition key in your bucket policy to deny any uploads where payloads are not signed. For example:</p> <ul style="list-style-type: none"> <li>Deny uploads that use presigned URLs. For more information, see <a href="#">Authenticating Requests: Using Query Parameters (AWS Signature Version 4) (p. 36)</a>.</li> <li>Deny uploads that use Authorization header to authenticate requests but don't sign the payload. For more information, see <a href="#">Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4) (p. 18)</a>.</li> </ul> <p>Valid value: UNSIGNED-PAYLOAD</p>

## Bucket Policy Examples Using Signature Version 4 Related Condition Keys

Deny any Amazon S3 action on the examplebucket to anyone if request is authenticated using Signature Version 4.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::examplebucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:signatureversion": "AWS4-HMAC-SHA256"
                }
            }
        }
    ]
}
```

The following bucket policy denies any Amazon S3 action on objects in examplebucket if the signature is more than ten minutes old.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Deny request if signature is more than 10 min old",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket3/*",  
            "Condition": {  
                "NumericGreaterThan": {  
                    "s3:signatureAge": 600000  
                }  
            }  
        }  
    ]  
}
```

The following bucket policy allows only requests that use the `Authorization` header for request authentication. Any POST or presigned URL requests will be denied.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Allow only requests that use Authorization header for request authentication. Deny POST or presigned URL requests.",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket3/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:authType": "REST-HEADER"  
                }  
            }  
        }  
    ]  
}
```

The following bucket policy denies any uploads that use presigned URLs.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Allow only requests that use Authorization header for request authentication. Deny POST or presigned URL requests.",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket3/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-content-sha256": "UNSIGNED-PAYLOAD"  
                }  
            }  
        }  
    ]  
}
```

# Authenticating Requests in Browser-Based Uploads Using POST (AWS Signature Version 4)

This section discusses how to upload files directly to Amazon S3 through a browser using HTTP POST requests. It also contains information about how to use the AWS Amplify JavaScript library for browser-based file uploads to Amazon S3.

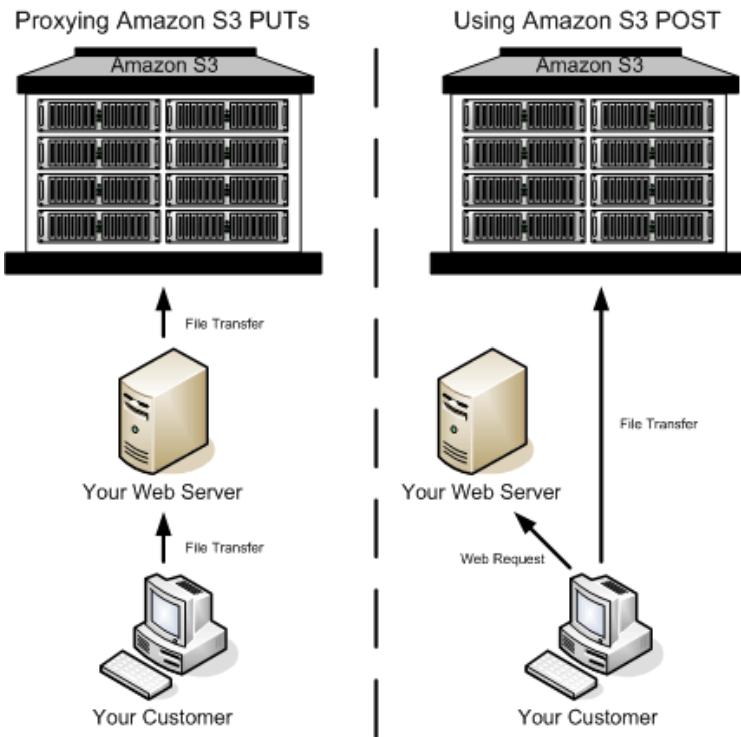
## Topics

- [Browser-Based Uploads Using HTTP POST \(p. 49\)](#)
- [Calculating a Signature \(p. 50\)](#)
- [Creating an HTML Form \(Using AWS Signature Version 4\) \(p. 51\)](#)
- [Creating a POST Policy \(p. 56\)](#)
- [Example: Browser-Based Upload using HTTP POST \(Using AWS Signature Version 4\) \(p. 61\)](#)
- [Using POST with Adobe Flash to Upload Objects \(p. 63\)](#)
- [Browser-Based Uploads to Amazon S3 Using the AWS Amplify Library \(p. 63\)](#)

## Browser-Based Uploads Using HTTP POST

Amazon S3 supports HTTP POST requests so that users can upload content directly to Amazon S3. By using POST, end users can authenticate requests without having to pass data through a secure intermediary node that protects your credentials. Thus, HTTP POST has the potential to reduce latency.

The following figure shows an Amazon S3 upload using a POST request.



### Uploading Using POST

1	The user accesses your page from a web browser.
2	Your webpage contains an HTTP form that contains all the information necessary for the user to upload content to Amazon S3.
3	The user uploads content to Amazon S3 through the web browser.

The process for sending browser-based POST requests is as follows:

1. Create a security policy specifying conditions that restrict what you want to allow in the request, such as the bucket name where objects can be uploaded, and key name prefixes that you want to allow for the object that is being created.
2. Create a signature that is based on the policy. For authenticated requests, the form must include a valid signature and the policy.
3. Create an HTML form that your users can access in order to upload objects to your Amazon S3 bucket.

The following section describes how to create a signature to authenticate a request. For information about creating forms and security policies, see [Creating an HTML Form \(Using AWS Signature Version 4\) \(p. 51\)](#).

## Calculating a Signature

For authenticated requests, the HTML form must include fields for a security policy and a signature.

- A security policy (see [Creating a POST Policy \(p. 56\)](#)) controls what is allowed in the request.

- The security policy is the `StringToSign` (see [Introduction to Signing Requests \(p. 15\)](#)) in your signature calculation.



### To Calculate a signature

- Create a policy using UTF-8 encoding.
- Convert the UTF-8-encoded policy bytes to base64. The result is the `StringToSign`.
- Create a signing key.
- Use the signing key to sign the `StringToSign` using HMAC-SHA256 signing algorithm.

For more information about creating HTML forms, security policies, and an example, see the following:

- [Creating an HTML Form \(Using AWS Signature Version 4\) \(p. 51\)](#)
- [Creating a POST Policy \(p. 56\)](#)
- [Example: Browser-Based Upload using HTTP POST \(Using AWS Signature Version 4\) \(p. 61\)](#)
- [Using POST with Adobe Flash to Upload Objects \(p. 63\)](#)

## Creating an HTML Form (Using AWS Signature Version 4)

### Topics

- [HTML Form Declaration \(p. 52\)](#)
- [HTML Form Fields \(p. 52\)](#)

To allow users to upload content to Amazon S3 by using their browsers (HTTP POST requests), you use HTML forms. HTML forms consist of a form declaration and form fields. The form declaration contains high-level information about the request. The form fields contain detailed request information.

This section describes how to create HTML forms. For a working example of browser-based upload using HTTP POST and related signature calculations for request authentication, see [Example: Browser-Based Upload using HTTP POST \(Using AWS Signature Version 4\) \(p. 61\)](#).

The form and policy must be UTF-8 encoded. You can apply UTF-8 encoding to the form by specifying `charset=UTF-8` in the `content` attribute. The following is an example of UTF-8 encoding in the HTML heading.

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

Following is an example of UTF-8 encoding in a request header.

```
Content-Type: text/html; charset=UTF-8
```

**Note**

The form data and boundaries (excluding the contents of the file) cannot exceed 20KB.

## HTML Form Declaration

The HTML form declaration has the following three attributes:

- `action` – The URL that processes the request, which must be set to the URL of the bucket. For example, if the name of your bucket is `examplebucket`, the URL is `http://examplebucket.s3.amazonaws.com/`.
- **Note**  
The key name is specified in a form field.
- `method` – The method must be `POST`.
- `enctype` – The enclosure type (`enctype`) must be set to `multipart/form-data` for both file uploads and text area uploads. For more information about `enctype`, see [RFC 1867](#).

This is a form declaration for the bucket `examplebucket`.

```
<form action="http://examplebucket.s3.amazonaws.com/" method="post"
      enctype="multipart/form-data">
```

## HTML Form Fields

The following table describes a list of fields that you can use within a form. Among other fields, there is a signature field that you can use to authenticate requests. There are fields for you to specify the signature calculation algorithm (`x-amz-algorithm`), the credential scope (`x-amz-credential`) that you used to generate the signing key, and the date (`x-amz-date`) used to calculate the signature. Amazon S3 uses this information to re-create the signature. If the signatures match, Amazon S3 processes the request.

**Note**

The variable  `${filename}`  is automatically replaced with the name of the file provided by the user and is recognized by all form fields. If the browser or client provides a full or partial path to the file, only the text following the last slash (/) or backslash (\) is used (for example, C:\Program Files\directory1\file.txt is interpreted as file.txt). If no file or file name is provided, the variable is replaced with an empty string.

If you don't provide elements required for authenticated requests, such as the `policy` element, the request is assumed to be anonymous and will succeed only if you have configured the bucket for public read and write.

Element Name	Description	Required
acl	<p>An Amazon S3 access control list (ACL). If an invalid ACL is specified, Amazon S3 denies the request. For more information about ACLs, see <a href="#">Using Amazon S3 ACLs</a>.</p> <p>Type: String</p> <p>Default: private</p> <p>Valid Values: <code>private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</code></p>	No
Cache-Control Content-Type Content-Disposition Content-Encoding Expires	REST-specific headers. For more information, see <a href="#">PUT Object (p. 412)</a> .	No
key	<p>The key name of the uploaded object.</p> <p>To use the file name provided by the user, use the <code> \${filename}</code> variable. For example, if you upload a file <code>photo1.jpg</code> and you specify <code>/user/user1/\${filename}</code> as key name, the file is stored as <code>/user/user1/photo1.jpg</code>.</p> <p>For more information, see <a href="#">Object Key and Metadata</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	Yes
policy	<p>The base64-encoded security policy that describes what is permitted in the request. For authenticated requests, a policy is required.</p> <p>Requests without a security policy are considered anonymous and will succeed only on a publicly writable bucket.</p>	Required for authenticated requests
success_action_redirect	<p>The URL to which the client is redirected upon successful upload.</p> <p>If <code>success_action_redirect</code> is not specified, or Amazon S3 cannot interpret the URL, Amazon S3 returns the empty document type that is specified in the <code>success_action_status</code> field.</p> <p>If the upload fails, Amazon S3 returns an error and does not redirect the user to another URL.</p>	No
success_action_status	The status code returned to the client upon successful upload if <code>success_action_redirect</code> is not specified.	No

Element Name	Description	Required
	<p>Valid values are 200, 201, or 204 (default).</p> <p>If the value is set to 200 or 204, Amazon S3 returns an empty document with the specified status code.</p> <p>If the value is set to 201, Amazon S3 returns an XML document with a 201 status code. For information about the content of the XML document, see <a href="#">POST Object (p. 385)</a>.</p> <p>If the value is not set or is invalid, Amazon S3 returns an empty document with a 204 status code.</p> <p><b>Note</b> Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting <code>success_action_status</code> to 201.</p>	
<code>x-amz-algorithm</code>	<p>The signing algorithm used to authenticate the request. For AWS Signature Version 4, the value is <code>AWS4-HMAC-SHA256</code>.</p> <p>This field is required if a policy document is included with the request.</p>	Required for authenticated requests
<code>x-amz-credential</code>	<p>In addition to your access key ID, this field also provides scope information identifying region and service for which the signature is valid. This should be the same scope you used in calculating the signing key for signature calculation.</p> <p>It is a string of the following form:</p> <p style="color: red;"><code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</code></p> <p>For example:</p> <p style="color: red;"><code>AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/aws4_request</code></p> <p>For Amazon S3, the <code>aws-service</code> string is <code>s3</code>. For a list of Amazon S3 <code>aws-region</code> strings, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>. This is required if a policy document is included with the request.</p>	Required for authenticated requests

Element Name	Description	Required
x-amz-date	<p>It is the date value in ISO8601 format. For example, 20130728T000000Z.</p> <p>It is the same date you used in creating the signing key (for example, 20130728). This must also be the same value you provide in the policy (x-amz-date) that you signed.</p> <p>This is required if a policy document is included with the request.</p>	Required for authenticated requests
x-amz-security-token	<p>A security token used by Amazon DevPay and session credentials</p> <p>If the request is using Amazon DevPay, it requires two x-amz-security-token form fields: one for the product token and one for the user token. For more information, see <a href="#">Using DevPay</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>If the request is using session credentials, it requires one x-amz-security-token form. For more information, see <a href="#">Requesting Temporary Security Credentials</a> in the <i>IAM User Guide</i>.</p>	No
x-amz-signature	<p>(AWS Signature Version 4) The HMAC-SHA256 hash of the security policy.</p> <p>This field is required if a policy document is included with the request.</p>	Required for authenticated requests
x-amz-meta-*	<p>Field names starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see <a href="#">PUT Object (p. 412)</a>.</p>	No
x-amz-*	See POST Object ( <a href="#">POST Object (p. 385)</a> ) for other x-amz-* headers.	No
file	<p>File or text content.</p> <p>The file or content must be the last field in the form.</p> <p>You cannot upload more than one file at a time.</p>	Yes

Conditional items are required for authenticated requests and are optional for anonymous requests.

Now that you know how to create forms, next you can create a security policy that you can sign. For more information, see [Creating a POST Policy \(p. 56\)](#).

# Creating a POST Policy

## Topics

- [Expiration \(p. 56\)](#)
- [Condition Matching \(p. 56\)](#)
- [Conditions \(p. 57\)](#)
- [Character Escaping \(p. 59\)](#)

The policy required for making authenticated requests using HTTP POST is a UTF-8 and base64-encoded document written in JavaScript Object Notation (JSON) that specifies conditions that the request must meet. Depending on how you design your policy document, you can control the access granularity per-upload, per-user, for all uploads, or according to other designs that meet your needs.

This section describes the POST policy. For example signature calculations using POST policy, see [Example: Browser-Based Upload using HTTP POST \(Using AWS Signature Version 4\) \(p. 61\)](#).

### Note

Although the policy document is optional, we highly recommend that you use one in order to control what is allowed in the request. If you make the bucket publicly writable, you have no control at all over which users can write to your bucket.

The following is an example of a POST policy document.

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"acl": "public-read" },
    {"bucket": "johnsmith" },
    ["starts-with", "$key", "user/eric/"],
  ]
}
```

The POST policy always contains the `expiration` and `conditions` elements. The example policy uses two condition matching types (exact matching and starts-with matching). The following sections describe these elements.

## Expiration

The `expiration` element specifies the expiration date and time of the POST policy in ISO8601 GMT date format. For example, `2013-08-01T12:00:00.000Z` specifies that the POST policy is not valid after midnight GMT on August 1, 2013.

## Condition Matching

Following is a table that describes condition matching types that you can use to specify POST policy conditions (described in the next section). Although you must specify one condition for each form field that you specify in the form, you can create more complex matching criteria by specifying multiple conditions for a form field.

Condition Match Type	Description
Exact Matches	The form field value must match the value specified. This example indicates that the ACL must be set to public-read:

Condition Match Type	Description
	<pre>{"acl": "public-read" }</pre> <p>This example is an alternate way to indicate that the ACL must be set to public-read:</p> <pre>[ "eq", "\$acl", "public-read" ]</pre>
Starts With	<p>The value must start with the specified value. This example indicates that the object key must start with user/user1:</p> <pre>[ "starts-with", "\$key", "user/user1/" ]</pre>
Matching Any Content	<p>To configure the POST policy to allow any content within a form field, use starts-with with an empty value (""). This example allows any value for success_action_redirect:</p> <pre>[ "starts-with", "\$success_action_redirect", "" ]</pre>
Specifying Ranges	<p>For form fields that accept a range, separate the upper and lower limit with a comma. This example allows a file size from 1 to 10 MiB:</p> <pre>[ "content-length-range", 1048579, 10485760 ]</pre>

The specific conditions supported in a POST policy are described in [Conditions \(p. 57\)](#).

## Conditions

The `conditions` in a POST policy is an array of objects, each of which is used to validate the request. You can use these conditions to restrict what is allowed in the request. For example, the preceding policy conditions require the following:

- Request must specify the `johnsmith` bucket name.
- Object key name must have the `user/eric` prefix.
- Object ACL must be set to `public-read`.

Each form field that you specify in a form (except `x-amz-signature`, `file`, `policy`, and field names that have an `x-ignore-` prefix) must appear in the list of conditions.

### Note

All variables within the form are expanded prior to validating the POST policy. Therefore, all condition matching should be against the expanded form fields. Suppose that you want to restrict your object key name to a specific prefix (`user/user1`). In this case, you set the key form field to `user/user1/${filename}`. Your POST policy should be `[ "starts-with", "$key", "user/user1/" ]` (do not enter `[ "starts-with", "$key", "user/user1/${filename}" ]`). For more information, see [Condition Matching \(p. 56\)](#).

Policy document conditions are described in the following table.

Element Name	Description
acl	<p>Specifies the ACL value that must be used in the form submission.</p> <p>This condition supports exact matching and <code>starts-with</code> condition match type discussed in the following section.</p>
bucket	<p>Specifies the acceptable bucket name.</p> <p>This condition supports exact matching condition match type.</p>
content-length-range	<p>The minimum and maximum allowable size for the uploaded content.</p> <p>This condition supports <code>content-length-range</code> condition match type.</p>
Cache-Control Content-Type Content-Disposition Content-Encoding Expires	<p>REST-specific headers. For more information, see <a href="#">POST Object (p. 385)</a>.</p> <p>This condition supports exact matching and <code>starts-with</code> condition match type.</p>
key	<p>The acceptable key name or a prefix of the uploaded object.</p> <p>This condition supports exact matching and <code>starts-with</code> condition match type.</p>
success_action_redirect redirect	<p>The URL to which the client is redirected upon successful upload.</p> <p>This condition supports exact matching and <code>starts-with</code> condition match type.</p>
success_action_status	<p>The status code returned to the client upon successful upload if <code>success_action_redirect</code> is not specified.</p> <p>This condition supports exact matching.</p>
x-amz-algorithm	<p>The signing algorithm that must be used during signature calculation. For AWS Signature Version 4, the value is <code>AWS4-HMAC-SHA256</code>.</p> <p>This condition supports exact matching.</p>
x-amz-credential	<p>The credentials that you used to calculate the signature. It provides access key ID and scope information identifying region and service for which the signature is valid. This should be the same scope you used in calculating the signing key for signature calculation.</p> <p>It is a string of the following form:</p> <p style="color: red;"><code>&lt;your-access-key-id&gt;/&lt;date&gt;/&lt;aws-region&gt;/&lt;aws-service&gt;/aws4_request</code></p>

Element Name	Description
	<p>For example:</p> <pre>AKIAIOSFODNN7EXAMPLE/20130728/us-east-1/s3/ aws4_request</pre> <p>For Amazon S3, the aws-service string is s3. For a list of Amazon S3 aws-region strings, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>. This is required if a POST policy document is included with the request.</p> <p>This condition supports exact matching.</p>
x-amz-date	<p>The date value specified in the ISO8601 formatted string. For example, 20130728T000000Z. The date must be same that you used in creating the signing key for signature calculation.</p> <p>This is required if a POST policy document is included with the request.</p> <p>This condition supports exact matching.</p>
x-amz-security-token	<p>Amazon DevPay security token.</p> <p>Each request that uses Amazon DevPay requires two x-amz-security-token form fields: one for the product token and one for the user token. As a result, the values must be separated by commas. For example, if the user token is eW91dHVIZQ== and the product token is b0hnNVNKWVJIQTA=, you set the POST policy entry to: { "x-amz-security-token": "eW91dHVIZQ==,b0hnNVNKWVJIQTA=" }.</p> <p>For more information about Amazon DevPay, see <a href="#">Using DevPay</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>
x-amz-meta-*	<p>User-specified metadata.</p> <p>This condition supports exact matching and starts-with condition match type.</p>
x-amz-*	<p>See POST Object (<a href="#">POST Object (p. 385)</a>) for other x-amz-* headers.</p> <p>This condition supports exact matching.</p>

#### Note

If your toolkit adds more form fields (for example, Flash adds `filename`), you must add them to the POST policy document. If you can control this functionality, prefix `x-ignore-` to the field so Amazon S3 ignores the feature and it won't affect future versions of this feature.

## Character Escaping

Characters that must be escaped within a POST policy document are described in the following table.

Escape Sequence	Description
\\	Backslash
\\$	Dollar symbol
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\uXXXX	All Unicode characters

Now that you are acquainted with forms and policies, and understand how signing works, you can try a POST upload example. You need to write the code to calculate the signature. The example provides a sample form, and a POST policy that you can use to test your signature calculations. For more information, see [Example: Browser-Based Upload using HTTP POST \(Using AWS Signature Version 4\) \(p. 61\)](#).

# Example: Browser-Based Upload using HTTP POST (Using AWS Signature Version 4)

This section shows an example of using an HTTP POST request to upload content directly to Amazon S3.

## Uploading a File to Amazon S3 Using HTTP POST

This example provides a sample POST policy and a form that you can use to upload a file. The topic uses the example policy and fictitious credentials to show you the workflow and resulting signature and policy hash. You can use this data as test suite to verify your signature calculation code.

The example uses the following example credentials the signature calculations. You can use these credentials to verify your signature calculation code. However, you must then replace these with your own credentials when sending requests to AWS.

Parameter	Value
AWSAccessKeyId	AKIAIOSFODNN7EXAMPLE
AWSSecretAccessKey	wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

## Sample Policy and Form

The following POST policy supports uploads to Amazon S3 with specific conditions.

```
{ "expiration": "2015-12-30T12:00:00.000Z",
  "conditions": [
    {"bucket": "sigv4examplebucket"},
    ["starts-with", "$key", "user/user1/"],
    {"acl": "public-read"},
    {"success_action_redirect": "http://sigv4examplebucket.s3.amazonaws.com/
successful_upload.html"},
    ["starts-with", "$Content-Type", "image/"],
    {"x-amz-meta-uuid": "14365123651274"},
    {"x-amz-server-side-encryption": "AES256"},
    ["starts-with", "$x-amz-meta-tag", ""],
    {"x-amz-credential": "AKIAIOSFODNN7EXAMPLE/20151229/us-east-1/s3/aws4_request"},
    {"x-amz-algorithm": "AWS4-HMAC-SHA256"},
    {"x-amz-date": "20151229T000000Z" }
  ]
}
```

This POST policy sets the following conditions on the request:

- The upload must occur before noon UTC on December 30, 2015.
- The content can be uploaded only to the `sigv4examplebucket`. The bucket must be in the region that you specified in the credential scope (`x-amz-credential` form parameter), because the signature you provided is valid only within this scope.
- You can provide any key name that starts with `user/user1`. For example, `user/user1/MyPhoto.jpg`.
- The ACL must be set to `public-read`.
- If the upload succeeds, the user's browser is redirected to `http://sigv4examplebucket.s3.amazonaws.com/successful_upload.html`.

- The object must be an image file.
- The `x-amz-meta-uuid` tag must be set to `14365123651274`.
- The `x-amz-meta-tag` can contain any value.

The following is a Base64-encoded version of this POST policy. You use this value as your `StringToSign` in signature calculation.

```
eyAiZXhwaxJhdGlvbiI6IC1yMDE1LTEyLTMwVDEyOjAwOjAwLjAwMFoILA0KICAiY29uZGl0aW9ucyI6IFsNCiAgICB7ImJ1Y2tldCI
```

When you copy/paste the preceding policy, it should only have newlines (not carriage return and new line) for your computed hash to match this value.

Using example credentials to create a signature, the signature value is as follows (in signature calculation, the date is same as the `x-amz-date` in the policy (20151229)):

```
8afdbf4008c03f22c2cd3cdb72e4afbb1f6a588f3255ac628749a66d7f09699e
```

The following example form specifies the preceding POST policy and supports a POST request to the `sigv4examplebucket`. Copy/paste the content in a text editor and save it as `exampleform.html`. You can then upload image files to the specific bucket using the `exampleform.html`. Your request will succeed if the signature you provide matches the signature Amazon S3 calculates.

#### Note

You must update the bucket name, dates, credential, policy, and signature with valid values for this to successfully upload to S3.

```
<html>
<head>

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

</head>
<body>

<form action="http://sigv4examplebucket.s3.amazonaws.com/" method="post"
enctype="multipart/form-data">
    Key to upload:
    <input type="input" name="key" value="user/user1/${filename}" /><br />
    <input type="hidden" name="acl" value="public-read" />
    <input type="hidden" name="success_action_redirect" value="http://
sigv4examplebucket.s3.amazonaws.com/successful_upload.html" />
    Content-Type:
    <input type="input" name="Content-Type" value="image/jpeg" /><br />
    <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
    <input type="hidden" name="x-amz-server-side-encryption" value="AES256" />
    <input type="text" name="X-Amz-Credential" value="AKIAIOSFODNN7EXAMPLE/20151229/us-
east-1/s3/aws4_request" />
    <input type="text" name="X-Amz-Algorithm" value="AWS4-HMAC-SHA256" />
    <input type="text" name="X-Amz-Date" value="20151229T000000Z" />

    Tags for File:
    <input type="input" name="x-amz-meta-tag" value="" /><br />
    <input type="hidden" name="Policy" value='<Base64-encoded policy string>' />
    <input type="hidden" name="X-Amz-Signature" value="<signature-value>" />
    File:
    <input type="file" name="file" /> <br />
    <!-- The elements after this will be ignored -->
    <input type="submit" name="submit" value="Upload to Amazon S3" />
</form>
```

```
</html>
```

The post parameters are case insensitive. For example, you can specify `x-amz-signature` or `X-Amz-Signature`.

## Using POST with Adobe Flash to Upload Objects

This section discusses uploading objects with an HTTP POST request when using Adobe Flash.

### Using POST with Adobe Flash

This section describes how to use POST with Adobe Flash.

#### Adobe Flash Player Security

By default, the Adobe Flash Player security model prohibits making network connections to servers outside the domain that serves the Adobe Flash (.swf) file.

To override the default, you must upload a publicly readable `crossdomain.xml` file to the bucket that will accept POST uploads. Here is a sample `crossdomain.xml` file:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

For more information about the Adobe Flash security model, go to the [Adobe web site](#).

When you add the `crossdomain.xml` file to your bucket, any Adobe Flash Player can connect to the `crossdomain.xml` file within your bucket. However, `crossdomain.xml` does not grant access to the Amazon S3 bucket.

#### Other Adobe Flash Considerations

The `FileReference` class in the Adobe Flash API adds the `Filename` form field to the POST request. When you build an Adobe Flash application that uploads files to Amazon S3 by using the `FileReference` class, include the following condition in your policy:

```
['starts-with', '$Filename', '' ]
```

Some versions of the Adobe Flash Player do not properly handle HTTP responses that have an empty body. To configure POST to return a response that does not have an empty body, set `success_action_status` to 201. Then, Amazon S3 will return an XML document with a 201 status code. For information about using this as an optional element (currently the only allowed value is the content of the XML document), see [POST Object \(p. 385\)](#). For information about form fields, see [HTML Form Fields \(p. 52\)](#).

## Browser-Based Uploads to Amazon S3 Using the AWS Amplify Library

This section describes how to upload files to Amazon S3 using the AWS Amplify JavaScript library.

For information about setting up the AWS Amplify library, see [AWS Amplify Installation and Configuration](#).

## Using the AWS Amplify JavaScript library to Upload Files to Amazon S3

The AWS Amplify library Storage module gives a simple browser-based upload mechanism for managing user content in public or private Amazon S3 storage.

### Example : AWS Amplify Manual Setup

The following example shows the manual setup for using the AWS Amplify Storage module. The default implementation of the Storage module uses Amazon S3.

```
import Amplify from 'aws-amplify';
Amplify.configure(
  Auth: {
    identityPoolId: 'XX-XXXX-X:XXXXXXXX-XXXX-1234-abcd-1234567890ab', //REQUIRED - Amazon Cognito Identity Pool ID
    region: 'XX-XXXX-X', // REQUIRED - Amazon Cognito Region
    userPoolId: 'XX-XXXX-X_abcd1234', //OPTIONAL - Amazon Cognito User Pool ID
    userPoolWebClientId: 'XX-XXXX-X_abcd1234', //OPTIONAL - Amazon Cognito Web Client ID
  },
  Storage: {
    bucket: '', //REQUIRED - Amazon S3 bucket
    region: 'XX-XXXX-X', //OPTIONAL - Amazon service region
  }
);
```

### Example : Put data into Amazon S3

The following example shows how to put public data into Amazon S3.

```
Storage.put('test.txt', 'Hello')
  .then(result => console.log(result))
  .catch(err => console.log(err));
```

The following example shows how to put private data into Amazon S3.

```
Storage.put('test.txt', 'Private Content', {
  level: 'private',
  contentType: 'text/plain'
})
  .then(result => console.log(result))
  .catch(err => console.log(err));
```

For more information about using the AWS Amplify Storage module, see [AWS Amplify Storage](#).

## More Info

[AWS Amplify Quick Start](#)

# Operations on the Service

This section describes operations you can perform on the Amazon S3 service.

## Topics

- [GET Service \(p. 65\)](#)

## GET Service

### Description

This implementation of the GET operation returns a list of all buckets owned by the authenticated sender of the request.

To authenticate a request, you must use a valid AWS Access Key ID that is registered with Amazon S3. Anonymous requests cannot list buckets, and you cannot list buckets that you did not create.

### Requests

#### Syntax

```
GET / HTTP/1.1
Host: s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

#### Request Parameters

This implementation of the operation does not use request parameters.

#### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

#### Request Elements

This implementation of the operation does not use request elements.

#### Responses

#### Response Elements

Name	Description
Bucket	Container for bucket information.  Type: Container

Name	Description
	<p>Children: Name, CreationDate</p> <p>Ancestor: ListAllMyBucketsResult.Buckets</p>
Buckets	<p>Container for one or more buckets.</p> <p>Type: Container</p> <p>Children: Bucket</p> <p>Ancestor: ListAllMyBucketsResult</p>
CreationDate	<p>Date the bucket was created.</p> <p>Type: date ( of the form yyyy-mm-ddThh:mm:ss.timezone, e.g., 2009-02-03T16:45:09.000Z)</p> <p>Ancestor: ListAllMyBucketsResult.Buckets.Bucket</p>
DisplayName	<p>Bucket owner's display name.</p> <p><b>Important</b> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p> <p>Type: String</p> <p>Ancestor: ListAllMyBucketsResult.Owner</p>
ID	<p>Bucket owner's canonical user ID.</p> <p>Type: String</p> <p>Ancestor: ListAllMyBucketsResult.Owner</p>
ListAllMyBucketsResult	<p>Container for response.</p> <p>Type: Container</p> <p>Children: Owner, Buckets</p> <p>Ancestor: None</p>
Name	<p>Bucket's name.</p> <p>Type: String</p> <p>Ancestor: ListAllMyBucketsResult.Buckets.Bucket</p>
Owner	<p>Container for bucket owner information.</p> <p>Type: Container</p> <p>Ancestor: ListAllMyBucketsResult</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The GET operation on the Service endpoint (`s3.amazonaws.com`) returns a list of all of the buckets owned by the authenticated sender of the request.

```
GET / HTTP/1.1
Host: s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Owner>
    <ID>bcaf1ffd86f461ca5fb16fd081034f</ID>
    <DisplayName>webfile</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>quotes</Name>
      <CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

## Related Resources

- [GET Bucket \(List Objects\) Version 1 \(p. 111\)](#)
- [GET Object \(p. 349\)](#)

# Operations on AWS Accounts

This section describes the REST operations related to Amazon S3 that you can perform on Amazon Web Services accounts.

## Topics

- [DELETE PublicAccessBlock \(p. 68\)](#)
- [GET PublicAccessBlock \(p. 69\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)

## DELETE PublicAccessBlock

### Description

This operation removes the `PublicAccessBlock` configuration for an Amazon Web Services account. In order to use this operation, you must have the `s3:PutAccountPublicAccessBlock` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

### Requests

#### Syntax

```
DELETE /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

#### Note

For information about locating your AWS account ID, see [Finding your AWS Account ID](#) in the *Amazon Web Services General Reference*.

### Request Parameters

This operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Related Resources

- [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service Developer Guide*.
- [GET PublicAccessBlock \(p. 153\)](#)
- [PUT PublicAccessBlock \(p. 277\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [GET PublicAccessBlock \(p. 69\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)

## GET PublicAccessBlock

### Description

This operation retrieves the `PublicAccessBlock` configuration for an Amazon Web Services account. In order to use this operation, you must have the `s3:GetAccountPublicAccessBlock` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

#### Important

When Amazon S3 evaluates the `PublicAccessBlock` configuration for a bucket or an object, it checks the `PublicAccessBlock` configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the `PublicAccessBlock` settings are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see [The Meaning of "Public"](#) in the *Amazon Simple Storage Service Developer Guide*.

### Requests

#### Syntax

```
GET /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
```

```
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

#### Note

For information about locating your AWS account ID, see [Finding your AWS Account ID](#) in the [Amazon Web Services General Reference](#).

## Request Parameters

This operation does not use request parameters.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
PublicAccessBlockConfiguration	Block configuration.  Type: Container  Children: BlockPublicAcls, IgnorePublicAccls, BlockPublicPolicy, RestrictPublicBuckets
BlockPublicAccls	Specifies whether Amazon S3 will block public access control lists (ACLs) for buckets and objects that are owned by this account.  Type: Boolean  Ancestor: PublicAccessBlockConfiguration  Valid values: TRUE   FALSE
IgnorePublicAccls	Specifies whether Amazon S3 will ignore public ACLs for buckets and objects that are owned by this account.  Type: Boolean  Ancestor: PublicAccessBlockConfiguration  Valid values: TRUE   FALSE

Name	Description
BlockPublicPolicy	<p>Specifies whether Amazon S3 will block public bucket policies for buckets that are owned by this account.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>
RestrictPublicBuckets	<p>Specifies whether Amazon S3 will restrict public bucket policies for buckets that are owned by this account.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request gets an account PublicAccessBlock configuration.

```
GET /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PublicAccessBlockConfiguration>
  <BlockPublicAccls>TRUE</BlockPublicAccls>
  <IgnorePublicAccls>FALSE</IgnorePublicAccls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

## Related Resources

- [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service Developer Guide*.

- [GET PublicAccessBlock \(p. 153\)](#)
- [PUT PublicAccessBlock \(p. 277\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)
- [DELETE PublicAccessBlock \(p. 68\)](#)

## PUT PublicAccessBlock

### Description

This operation creates or modifies the `PublicAccessBlock` configuration for an Amazon Web Services account. In order to use this operation, you must have the `s3:PutAccountPublicAccessBlock` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy in the Amazon Simple Storage Service Developer Guide](#).

#### Important

When Amazon S3 evaluates the `PublicAccessBlock` configuration for a bucket or an object, it checks the `PublicAccessBlock` configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the `PublicAccessBlock` configurations are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or object public, see [The Meaning of "Public" in the Amazon Simple Storage Service Developer Guide](#).

### Requests

#### Syntax

```
PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

#### Note

For information about locating your AWS account ID, see [Finding your AWS Account ID in the Amazon Web Services General Reference](#).

### Request Parameters

This operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This operation uses the following request elements. You can enable `BlockPublicAcls`, `IgnorePublicAcls`, `BlockPublicPolicy`, and `RestrictPublicBuckets` in any combination.

Name	Description	Required
PublicAccessBlockConfiguration	<p><b>Block configuration.</b> You can enable the configuration elements in any combination.</p> <p>Type: Container</p> <p>Children: <code>BlockPublicAcls</code>, <code>IgnorePublicAccls</code>, <code>BlockPublicPolicy</code>, <code>RestrictPublicBuckets</code></p>	Yes
BlockPublicAcls	<p>Specifies whether Amazon S3 should block public access control lists (ACLs) for buckets and objects in this account. Setting this element to TRUE causes the following behavior:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Bucket acl (p. 235)</a> and <a href="#">PUT Object acl (p. 447)</a> calls fail if the specified ACL is public.</li> <li>• <a href="#">PUT Object (p. 412)</a> calls fail if the request includes a public ACL.</li> <li>• <a href="#">PUT Bucket (p. 227)</a> calls fail if the request includes a public ACL.</li> </ul> <p><b>Important</b> Enabling this setting doesn't affect existing policies or ACLs.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: TRUE   FALSE</p>	No
IgnorePublicAccls	<p>Specifies whether Amazon S3 should ignore public ACLs for buckets and objects in this account. Setting this element to TRUE causes Amazon S3 to ignore all public ACLs on buckets in this account and objects in those buckets.</p> <p><b>Important</b> Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: TRUE   FALSE</p>	No
BlockPublicPolicy	<p>Specifies whether Amazon S3 should block public bucket policies for buckets in this account. Setting this element to TRUE causes Amazon S3 to reject calls to <a href="#">PUT Bucket policy (p. 300)</a> if the specified bucket policy allows public access.</p> <p><b>Important</b> Enabling this setting doesn't affect existing bucket policies.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: TRUE   FALSE</p>	No

Name	Description	Required
RestrictPublicBuckets	<p><b>Specifies</b> whether Amazon S3 should restrict public bucket policies for buckets in this account. If this element is set to TRUE, then only AWS services and authorized users within this account can access buckets with public policies.</p> <p><b>Important</b> Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>	No

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### First Sample Request

The following request puts an account PublicAccessBlock configuration that blocks public ACLs for buckets in the specified account.

```

PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
    <BlockPublicAcls>TRUE</BlockPublicAcls>
    <IgnorePublicAcls>FALSE</IgnorePublicAcls>
    <BlockPublicPolicy>FALSE</BlockPublicPolicy>
    <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>

```

## First Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0
```

## Second Sample Request

The following request puts an account PublicAccessBlock configuration that ignores public ACLs and restricts public bucket policies for buckets in the specified account.

```
PUT /v20180820/configuration/publicAccessBlock HTTP/1.1
Host: <account-id>.s3-control.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
    <BlockPublicAcls>FALSE</BlockPublicAcls>
    <IgnorePublicAcls>TRUE</IgnorePublicAcls>
    <BlockPublicPolicy>FALSE</BlockPublicPolicy>
    <RestrictPublicBuckets>TRUE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

## Second Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0
```

## Related Resources

- [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service Developer Guide*.
- [GET PublicAccessBlock \(p. 153\)](#)
- [PUT PublicAccessBlock \(p. 277\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [GET PublicAccessBlock \(p. 69\)](#)
- [DELETE PublicAccessBlock \(p. 68\)](#)

# Operations on Buckets

This section describes operations you can perform on Amazon S3 buckets.

## Topics

- [DELETE Bucket \(p. 78\)](#)
- [DELETE Bucket analytics \(p. 80\)](#)
- [DELETE Bucket cors \(p. 82\)](#)
- [DELETE Bucket encryption \(p. 84\)](#)
- [DELETE Bucket inventory \(p. 86\)](#)
- [DELETE Bucket lifecycle \(p. 88\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)
- [DELETE Bucket metrics \(p. 90\)](#)
- [DELETE Bucket policy \(p. 93\)](#)
- [DELETE Bucket replication \(p. 95\)](#)
- [DELETE Bucket tagging \(p. 97\)](#)
- [DELETE Bucket website \(p. 99\)](#)
- [GET Bucket \(List Objects\) Version 2 \(p. 101\)](#)
- [GET Bucket accelerate \(p. 120\)](#)
- [GET Bucket acl \(p. 123\)](#)
- [GET Bucket analytics \(p. 126\)](#)
- [GET Bucket cors \(p. 131\)](#)
- [GET Bucket encryption \(p. 135\)](#)
- [GET Bucket Inventory \(p. 139\)](#)
- [GET Bucket lifecycle \(p. 145\)](#)
- [GET Bucket location \(p. 152\)](#)
- [GET PublicAccessBlock \(p. 153\)](#)
- [GET Bucket logging \(p. 157\)](#)
- [GET Bucket metrics \(p. 160\)](#)
- [GET Bucket notification \(p. 164\)](#)
- [GET Bucket object lock configuration \(p. 169\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [GET Bucket Object versions \(p. 173\)](#)
- [GET Bucket policy \(p. 185\)](#)
- [GET Bucket replication \(p. 187\)](#)
- [GET Bucket requestPayment \(p. 194\)](#)
- [GET Bucket tagging \(p. 196\)](#)
- [GET Bucket versioning \(p. 199\)](#)
- [GET Bucket website \(p. 202\)](#)
- [HEAD Bucket \(p. 204\)](#)
- [List Bucket Analytics Configurations \(p. 206\)](#)
- [List Bucket Inventory Configurations \(p. 210\)](#)
- [List Bucket Metrics Configurations \(p. 215\)](#)
- [List Multipart Uploads \(p. 218\)](#)

- [PUT Bucket \(p. 227\)](#)
- [PUT Bucket accelerate \(p. 232\)](#)
- [PUT Bucket acl \(p. 235\)](#)
- [PUT Bucket analytics \(p. 242\)](#)
- [PUT Bucket cors \(p. 248\)](#)
- [PUT Bucket encryption \(p. 254\)](#)
- [PUT Bucket inventory \(p. 258\)](#)
- [PUT Bucket lifecycle \(p. 265\)](#)
- [PUT PublicAccessBlock \(p. 277\)](#)
- [PUT Bucket logging \(p. 281\)](#)
- [PUT Bucket metrics \(p. 285\)](#)
- [PUT Bucket notification \(p. 290\)](#)
- [PUT Bucket object lock configuration \(p. 298\)](#)
- [PUT Bucket policy \(p. 300\)](#)
- [PUT Bucket replication \(p. 302\)](#)
- [PUT Bucket requestPayment \(p. 312\)](#)
- [PUT Bucket tagging \(p. 314\)](#)
- [PUT Bucket versioning \(p. 317\)](#)
- [PUT Bucket website \(p. 321\)](#)
- [DefaultRetention \(p. 330\)](#)
- [ObjectLockConfiguration \(p. 331\)](#)
- [ObjectLockRule \(p. 332\)](#)

# DELETE Bucket

## Description

Deletes the bucket named in the URI. All objects (including all object versions and delete markers) in the bucket must be deleted before the bucket itself can be deleted.

## Requests

### Syntax

```
DELETE / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request deletes the bucket named "quotes".

```
DELETE / HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

## Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagkKdBX2z7I1/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Wed, 01 Mar 2006 12:00:00 GMT
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 227\)](#)
- [DELETE Object \(p. 343\)](#)

# DELETE Bucket analytics

## Description

This implementation of the `DELETE` operation deletes an analytics configuration (identified by the analytics configuration ID) from the bucket.

To use this operation, you must have permissions to perform the `s3:PutAnalyticsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about Amazon S3 analytics feature, see [Amazon S3 Analytics – Storage Class Analysis](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?analytics&id=analytics-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of `DELETE` uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID that identifies the analytics configuration.  Type: String  Default: None  Valid Characters for <code>id</code> : a-z A-Z 0-9 - _ .	Yes

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

# Responses

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Sample Request

The following DELETE request deletes the analytics configuration with the ID list1.

```
DELETE ?/analytics&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 May 2014 02:11:22 GMT
Authorization: signatureValue
```

### Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The analytics configuration with the ID list1 for the bucket has been removed.

```
HTTP/1.1 204 No Content
x-amz-id-2: OFmFIWsh/PpBuzzZ0JFRC55ZGVmQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1B0l5
x-amz-request-id: OCF038E9BCF63097
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AmazonS3
```

## Related Resources

- [GET Bucket analytics \(p. 126\)](#)
- [List Bucket Analytics Configurations \(p. 206\)](#)
- [PUT Bucket analytics \(p. 242\)](#)

# DELETE Bucket cors

## Description

Deletes the `cors` configuration information set for the bucket.

To use this operation, you must have permission to perform the `s3:PutBucketCORS` action. The bucket owner has this permission by default and can grant this permission to others.

For information more about `cors`, go to [Enabling Cross-Origin Resource Sharing](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?cors HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Example 1: Retrieve cors subresource

The following DELETE request deletes the `cors` subresource from the specified bucket. This action removes `cors` configuration that is stored in the subresource.

#### Sample Request

```
DELETE /?cors HTTP/1.1
```

```
Host: examplebucket.s3.amazonaws.com
Date: Tue, 13 Dec 2011 19:14:42 GMT
Authorization: signatureValue
```

## Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: OFmFIWsh/PpBuzzOJFRC55ZGVmQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1B0l5
x-amz-request-id: OCF038E9BCF63097
Date: Tue, 13 Dec 2011 19:14:42 GMT
Server: AmazonS3
Content-Length: 0
```

## Related Resources

- [PUT Bucket cors \(p. 248\)](#)
- [DELETE Bucket cors \(p. 82\)](#)
- [OPTIONS object \(p. 382\)](#)

# DELETE Bucket encryption

## Description

This implementation of the `DELETE` operation removes default encryption from the bucket. For information about the Amazon S3 default encryption feature, see [Amazon S3 Default Bucket Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

To use this operation, you must have permissions to perform the `s3:PutEncryptionConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?encryption HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Sample Request

The following `DELETE` request deletes default encryption from the bucket.

```
DELETE /?encryption HTTP/1.1
```

```
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: signatureValue
```

## Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response confirming that default encryption has been removed from the bucket.

```
HTTP/1.1 204 No Content
x-amz-id-2: OFmFIWsh/PpBuzz0JFRC55ZGVmQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1Bol5
x-amz-request-id: OCF038E9BCF63097
Date: Wed, 06 Sep 2017 12:00:00 GMT
Server: AmazonS3
```

## Related Resources

- [GET Bucket encryption \(p. 135\)](#)
- [PUT Bucket encryption \(p. 254\)](#)

# DELETE Bucket inventory

## Description

This implementation of the `DELETE` operation deletes an inventory configuration (identified by the inventory configuration ID) from the bucket.

To use this operation, you must have permissions to perform the `s3:PutInventoryConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about the Amazon S3 inventory feature, see [Amazon S3 Inventory](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of `DELETE` uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID that identifies the inventory configuration.  Type: String  Default: None  Valid Characters for <code>id</code> : a-z A-Z 0-9 - _ .	Yes

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

# Responses

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Sample Request

The following DELETE request deletes the inventory configuration with the ID list1.

```
DELETE ?/inventory&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 May 2014 02:11:22 GMT
Authorization: signatureValue
```

### Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The inventory configuration with the ID list1 for the bucket has been removed.

```
HTTP/1.1 204 No Content
x-amz-id-2: OFmFIWsh/PpBuzzZ0JFRC55ZGVmQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1B0l5
x-amz-request-id: OCF038E9BCF63097
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AmazonS3
```

## Related Resources

- [GET Bucket Inventory \(p. 139\)](#)
- [List Bucket Inventory Configurations \(p. 210\)](#)
- [PUT Bucket inventory \(p. 258\)](#)

# DELETE Bucket lifecycle

## Description

Deletes the lifecycle configuration from the specified bucket. Amazon S3 removes all the lifecycle configuration rules in the lifecycle subresource associated with the bucket. Your objects never expire, and Amazon S3 no longer automatically deletes any objects on the basis of rules contained in the deleted lifecycle configuration.

To use this operation, you must have permission to perform the `s3:PutLifecycleConfiguration` action. By default, the bucket owner has this permission and the bucket owner can grant this permission to others.

There is usually some time lag before lifecycle configuration deletion is fully propagated to all the Amazon S3 systems.

For more information about the object expiration, go to [Elements to Describe Lifecycle Actions](#) in the [Amazon Simple Storage Service Developer Guide](#).

## Requests

### Syntax

```
DELETE /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Sample Request

The following DELETE request deletes the `lifecycle` subresource from the specified bucket. This removes lifecycle configuration stored in the subresource.

```
DELETE /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 Dec 2011 05:37:16 GMT
Authorization: signatureValue
```

### Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. Objects in your bucket no longer expire.

```
HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLEtRPfTaOAa==
x-amz-request-id: 656c76696e672SAMPLE5657374
Date: Wed, 14 Dec 2011 05:37:16 GMT
Connection: keep-alive
Server: AmazonS3
```

## Related Resources

- [PUT Bucket lifecycle \(p. 265\)](#)
- [GET Bucket lifecycle \(p. 145\)](#)

## DELETE PublicAccessBlock

### Description

This operation removes the `PublicAccessBlock` configuration for an Amazon S3 bucket. In order to use this operation, you must have the `s3:PutBucketPublicAccessBlock` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

### Requests

#### Syntax

```
DELETE /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

#### Request Parameters

This operation does not use request parameters.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Related Resources

- Using Amazon S3 Block Public Access in the *Amazon Simple Storage Service Developer Guide*.
- [GET PublicAccessBlock \(p. 153\)](#)
- [PUT PublicAccessBlock \(p. 277\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [GET PublicAccessBlock \(p. 69\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)
- [DELETE PublicAccessBlock \(p. 68\)](#)

# DELETE Bucket metrics

## Description

Deletes a metrics configuration for the Amazon CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. Note that this doesn't include the daily storage metrics.

To use this operation, you must have permissions to perform the `s3:PutMetricsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about CloudWatch request metrics for Amazon S3, see [Monitoring Metrics with Amazon CloudWatch](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?metrics&id=Id HTTP/1.1
HOST: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

Parameter	Description	Required
<code>id</code>	The ID used to identify the metrics configuration.	Yes

### Request Headers

This operation uses only Request Headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This operation does not use request elements.

### Responses

#### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

#### Response Elements

This implementation of the operation does not return response elements.

#### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the `ExampleMetrics` value for the `FilterId` dimension.

```
DELETE /?metrics&id=ExampleMetrics HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
```

Authorization: *signatureValue*

## Sample Response

Delete the metric configuration with a specified ID, which disables the CloudWatch metrics with the ExampleMetrics value for the FilterId dimension.

```
HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
```

## Related Resources

- [GET Bucket metrics \(p. 160\)](#)
- [PUT Bucket metrics \(p. 285\)](#)
- [List Bucket Metrics Configurations \(p. 215\)](#)
- [Monitoring Metrics with Amazon CloudWatch in the \*Amazon Simple Storage Service Developer Guide\*.](#)

# DELETE Bucket policy

## Description

This implementation of the `DELETE` operation uses the `policy` subresource to return the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the `DeleteBucketPolicy` permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have `DeleteBucketPolicy` permissions, Amazon S3 returns a `403 Access Denied` error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a `405 Method Not Allowed` error.

### Important

As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see [Using Bucket Policies and User Policies](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

The response elements contain the status of the `DELETE` operation including the error code if the request failed.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request deletes the bucket named `BucketName`.

```
DELETE /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Tue, 04 Apr 2010 20:34:56 GMT
Authorization: signatureValue
```

### Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLEtRPfTaOFg==
x-amz-request-id: 656c76696e672SAMPLE5657374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 227\)](#)
- [DELETE Object \(p. 343\)](#)

# DELETE Bucket replication

## Description

Deletes the `replication` subresource associated with the specified bucket. This deletes the replication configuration from the bucket.

To use this operation, you must have permissions to perform the `s3:PutReplicationConfiguration` action. The bucket owner has these permissions by default and can grant it to others. For information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

### Note

It can take a while for the deletion of a replication configuration to fully propagate.

For information about replication configuration, see [Cross-Region Replication](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?replication HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string
```

For more information about authorization, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

The following DELETE request deletes the `replication` subresource from the specified bucket. This removes the replication configuration that is set for the bucket.

```
DELETE /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 11 Feb 2015 05:37:16 GMT
20150211T171320Z

Authorization: authorization string
```

When the replication subresource has been deleted, Amazon S3 returns a 204 No Content response. It will not replicate new objects that are stored in the examplebucket bucket.

```
HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByRx9e6j5OnimrSAMPLEtRPfTaOAa==
x-amz-request-id: 656c76696e672example
Date: Wed, 11 Feb 2015 05:37:16 GMT
Connection: keep-alive
Server: AmazonS3
```

## Related Resources

- [PUT Bucket replication \(p. 302\)](#)
- [GET Bucket replication \(p. 187\)](#)

# DELETE Bucket tagging

## Description

This implementation of the `DELETE` operation uses the `tagging` subresource to remove a tag set from the specified bucket.

To use this operation, you must have permission to perform the `s3:PutBucketTagging` action. By default, the bucket owner has this permission and can grant this permission to others.

## Requests

### Syntax

```
DELETE /?tagging HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Sample Request

The following `DELETE` request deletes the tag set from the specified bucket.

```
DELETE /?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 14 Dec 2011 05:37:16 GMT
Authorization: signatureValue
```

## Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The tag set for the bucket has been removed.

```
HTTP/1.1 204 No Content
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3
```

## Related Resources

- [GET Bucket tagging \(p. 196\)](#)
- [PUT Bucket tagging \(p. 314\)](#)

# DELETE Bucket website

## Description

This operation removes the website configuration for a bucket. Amazon S3 returns a 200 OK response upon successfully deleting a website configuration on the specified bucket. You will get a 200 OK response if the website configuration you are trying to delete does not exist on the bucket. Amazon S3 returns a 404 response if the bucket specified in the request does not exist.

This DELETE operation requires the S3:DeleteBucketWebsite permission. By default, only the bucket owner can delete the website configuration attached to a bucket. However, bucket owners can grant other users permission to delete the website configuration by writing a bucket policy granting them the S3:DeleteBucketWebsite permission.

For more information about hosting websites, go to [Hosting Websites on Amazon S3](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Examples

### Sample Request

This request deletes the website configuration on the specified bucket.

```
DELETE ?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue
```

### Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: aws-s3integ-s3ws-31008.sea31.amazonaws.com
x-amz-request-id: AF1DD829D3B49707
Date: Thu, 03 Feb 2011 22:10:26 GMT
Server: AmazonS3
```

### Related Resources

- [GET Bucket website \(p. 202\)](#)
- [PUT Bucket website \(p. 321\)](#)

# GET Bucket (List Objects) Version 2

## Description

This implementation of the GET operation returns some or all (up to 1,000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A 200 OK response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

To use this implementation of the operation, you must have READ access to the bucket.

To use this operation in an AWS Identity and Access Management (IAM) policy, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the [Amazon Simple Storage Service Developer Guide](#).

### Important

This section describes the latest revision of the API. We recommend that you use this revised API, GET Bucket (List Objects) version 2, for application development. For backward compatibility, Amazon S3 continues to support the prior version of this API, GET Bucket (List Objects) version 1. For more information about the previous version, see [GET Bucket \(List Objects\) Version 1 \(p. 111\)](#).

### Note

To get a list of your buckets, see [GET Service \(p. 65\)](#).

## Requests

### Syntax

```
GET /?list-type=2 HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of GET uses the parameters in the following table.

Parameter	Description	Required
delimiter	A delimiter is a character you use to group keys.  If you specify a prefix, all of the keys that contain the same string between the prefix and the first occurrence of the delimiter after the prefix are grouped under a single result element called CommonPrefixes. If you don't specify the prefix parameter, the substring starts at the beginning of the key. The keys that are grouped under the CommonPrefixes result element are not returned elsewhere in the response.  Type: String  Default: None	No

Parameter	Description	Required
encoding-type	<p>Requests Amazon S3 to encode the response and specifies the encoding method to use.</p> <p>An object key can contain any Unicode character. However, XML 1.0 parsers cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid value: url</p>	No
max-keys	<p>Sets the maximum number of keys returned in the response body. If you want to retrieve fewer than the default 1,000 keys, you can add this to your request.</p> <p>The response might contain fewer keys, but it never contains more. If there are additional keys that satisfy the search criteria, but these keys were not returned because max-keys was exceeded, the response contains &lt;IsTruncated&gt;true&lt;/IsTruncated&gt;. To return the additional keys, see NextContinuationToken.</p> <p>Type: String</p> <p>Default: 1000</p>	No
prefix	<p>Limits the response to keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.)</p> <p>Type: String</p> <p>Default: None</p>	No
list-type	<p>Version 2 of the API requires this parameter and you must set its value to 2.</p> <p>Type: String</p> <p>Default: The value is always 2.</p>	Yes
continuation-token	<p>When the response to this API call is truncated (that is, the IsTruncated response element value is true), the response also includes the NextContinuationToken element. To list the next set of objects, you can use the NextContinuationToken element in the next request as the continuation-token.</p> <ul style="list-style-type: none"> <li>• The continuation token is an opaque value that Amazon S3 understands.</li> <li>• Amazon S3 lists objects in UTF-8 character encoding in lexicographical order.</li> </ul> <p>Type: String</p> <p>Default: None</p>	No

Parameter	Description	Required
<code>fetch-owner</code>	<p>By default, the API does not return the <code>Owner</code> information in the response. If you want the owner information in the response, you can specify this parameter with the value set to true.</p> <p>Type: String</p> <p>Default: false</p>	No
<code>start-after</code>	<p>If you want the API to return key names after a specific object key in your key space, you can add this parameter. Amazon S3 lists objects in UTF-8 character encoding in lexicographical order.</p> <p>This parameter is valid only in your first request. If the response is truncated, you can specify this parameter along with the <code>continuation-token</code> parameter, and then Amazon S3 ignores this parameter.</p> <p>Type: String</p> <p>Default: None</p>	No

## Request Elements

This implementation of the operation does not use request elements.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
<code>Contents</code>	<p>Metadata about each object returned.</p> <p>Type: XML metadata</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>CommonPrefixes</code>	<p>All of the keys rolled up into a common prefix count as a single return when calculating the number of returns. See <code>MaxKeys</code>.</p> <ul style="list-style-type: none"> <li>A response can contain <code>CommonPrefixes</code> only if you specify a delimiter.</li> <li><code>CommonPrefixes</code> contains all (if there are any) keys between <code>Prefix</code> and the next occurrence of the string specified by a delimiter.</li> </ul>

Name	Description
	<ul style="list-style-type: none"> <li>• <code>CommonPrefixes</code> lists keys that act like subdirectories in the directory specified by <code>Prefix</code>.</li> </ul> <p>For example, if the prefix is <code>notes/</code> and the delimiter is a slash (<code>/</code>) as in <code>notes/summer/july</code>, the common prefix is <code>notes/summer/</code>. All of the keys that roll up into a common prefix count as a single return when calculating the number of returns. See <code>MaxKeys</code>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>Delimiter</code>	<p>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the <code>CommonPrefixes</code> collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the <code>MaxKeys</code> value.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>DisplayName</code>	<p>Object owner's name.</p> <p><b>Important</b> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents.Owner</code></p>
<code>Encoding-Type</code>	<p>Encoding type used by Amazon S3 to encode object key names in the XML response.</p> <p>If you specify <code>encoding-type</code> request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements:</p> <p><code>Delimiter</code>, <code>Prefix</code>, <code>Key</code>, and <code>StartAfter</code>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>ETag</code>	<p>The entity tag is an MD5 hash of the object. <code>ETag</code> reflects only changes to the contents of an object, not its metadata.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>

Name	Description
<code>ID</code>	<p>Object owner's ID.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents.Owner</code></p>
<code>IsTruncated</code>	<p>Set to <code>false</code> if all of the results were returned. Set to <code>true</code> if more keys are available to return. If the number of results exceeds that specified by <code>MaxKeys</code>, all of the results might not be returned.</p> <p>Type: Boolean</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>Key</code>	<p>The object's key.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>
<code>LastModified</code>	<p>Date and time the object was last modified.</p> <p>Type: Date</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>
<code>MaxKeys</code>	<p>The maximum number of keys returned in the response body.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>Name</code>	<p>Name of the bucket.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>Owner</code>	<p>Bucket owner.</p> <p>Type: String</p> <p>Children: <code>DisplayName, ID</code></p> <p>Ancestor: <code>ListBucketResult.Contents   CommonPrefixes</code></p>
<code>Prefix</code>	<p>Keys that begin with the indicated prefix.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>Size</code>	<p>Size in bytes of the object.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>

Name	Description
StorageClass	<p>STANDARD   STANDARD_IA   ONEZONE_IA   REDUCED_REDUNDANCY   GLACIER</p> <p>Type: String</p> <p>Ancestor: ListBucketResult.Contents</p>
ContinuationToken	<p>If ContinuationToken was sent with the request, it is included in the response.</p> <p>Type: String</p> <p>Ancestor: ListBucketResult</p>
KeyCount	<p>Returns the number of keys included in the response. The value is always less than or equal to the MaxKeys value.</p> <p>Type: String</p> <p>Ancestor: ListBucketResult</p>
NextContinuationToken	<p>If the response is truncated, Amazon S3 returns this parameter with a continuation token. You can specify the token as the continuation-token in your next request to retrieve the next set of keys.</p> <p>Type: String</p> <p>Ancestor: ListBucketResult</p>
StartAfter	<p>If StartAfter was sent with the request, it is included in the response.</p> <p>Type: String</p> <p>Ancestor: ListBucketResult</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Listing Keys

This request returns the objects in BucketName. The request specifies the list-type parameter, which indicates version 2 of the API.

#### Sample Request

```
GET /?list-type=2 HTTP/1.1
Host: bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
Content-Type: text/plain
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>bucket</Name>
  <Prefix/>
  <KeyCount>205</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>my-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    ...
  </Contents>
  ...
</ListBucketResult>
```

## Example 2: Listing Keys Using the max-keys, prefix, and start-after Parameters

In addition to the `list-type` parameter that indicates version 2 of the API, the request also specifies additional parameters to retrieve up to three keys in the `quotes` bucket that start with `E` and occur lexicographically after `ExampleGuide.pdf`.

## Sample Request

```
GET /?list-type=2&max-keys=3&prefix=E&start-after=ExampleGuide.pdf HTTP/1.1
Host: quotes.s3.amazonaws.com
x-amz-date: 20160430T232933Z
Authorization: authorization string
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>quotes</Name>
  <Prefix>E</Prefix>
  <StartAfter>ExampleGuide.pdf</StartAfter>
  <KeyCount>1</KeyCount>
  <MaxKeys>3</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>ExampleObject.txt</Key>
    <LastModified>2013-09-17T18:07:53.000Z</LastModified>
    <ETag>"599bab3ed2c697f1d26842727561fd94"</ETag>
    <Size>857</Size>
  </Contents>
</ListBucketResult>
```

```
<StorageClass>REDUCED_REDUNDANCY</StorageClass>
</Contents>
</ListBucketResult>
```

## Example 3: Listing Keys Using the prefix and delimiter Parameters

This example illustrates the use of the `prefix` and the `delimiter` parameters in the request. For this example, we assume that you have the following keys in your bucket:

```
sample.jpg
photos/2006/January/sample.jpg
photos/2006/February/sample2.jpg
photos/2006/February/sample3.jpg
photos/2006/February/sample4.jpg
```

The following GET request specifies the `delimiter` parameter with value `/`.

```
GET /?list-type=2&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T235931Z
Authorization: authorization string
```

The key `sample.jpg` does not contain the delimiter character, and Amazon S3 returns it in the `Contents` element in the response. However, all other keys contain the delimiter character. Amazon S3 groups these keys and returns a single `CommonPrefixes` element with the `prefix` value `photos/`. The element is a substring that starts at the beginning of these keys and ends at the first occurrence of the specified delimiter.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>example-bucket</Name>
<Prefix></Prefix>
<KeyCount>2</KeyCount>
<MaxKeys>1000</MaxKeys>
<Delimiter></Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
  <Key>sample.jpg</Key>
  <LastModified>2011-02-26T01:56:20.000Z</LastModified>
  <ETag>"bf1d737a4d46a19f3bcd6905cc8b902"</ETag>
  <Size>142863</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<CommonPrefixes>
  <Prefix>photos/</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

The following GET request specifies the `delimiter` parameter with value `/`, and the `prefix` parameter with value `photos/2006/`.

```
GET /?list-type=2&prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160501T000433Z
```

Authorization: *authorization string*

In response, Amazon S3 returns only the keys that start with the specified prefix. Further, it uses the delimiter character to group keys that contain the same substring until the first occurrence of the delimiter character after the specified prefix. For each such key group Amazon S3 returns one CommonPrefixes element in the response. The keys grouped under this CommonPrefixes element are not returned elsewhere in the response. The value returned in the CommonPrefixes element is a substring that starts at the beginning of the key and ends at the first occurrence of the specified delimiter after the prefix.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>example-bucket</Name>
<Prefix>photos/2006/</Prefix>
<KeyCount>3</KeyCount>
<MaxKeys>1000</MaxKeys>
<Delimiter>/</Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
  <Key>photos/2006/</Key>
  <LastModified>2016-04-30T23:51:29.000Z</LastModified>
  <ETag>&quot;d41d8cd98f00b204e9800998ecf8427e&quot;</ETag>
  <Size>0</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>

<CommonPrefixes>
  <Prefix>photos/2006/February/</Prefix>
</CommonPrefixes>
<CommonPrefixes>
  <Prefix>photos/2006/January/</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

## Example 4: Using a Continuation Token

In this example, the initial request returns more than 1,000 keys. In response to this request, Amazon S3 returns the IsTruncated element with the value set to true and with a NextContinuationToken element.

```
GET /?list-type=2 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 02 May 2016 23:17:07 GMT
Authorization: authorization string
```

The following is a sample response:

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXR3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>bucket</Name>
<Prefix></Prefix>
<NextContinuationToken>1ueGcxLPRx1Tr/XYExHnhbYLgveDs2J/wm36Hy4vb0wM=</
NextContinuationToken>
```

```
<KeyCount>1000</KeyCount>
<MaxKeys>1000</MaxKeys>
<IsTruncated>true</IsTruncated>
<Contents>
  <Key>happyface.jpg</Key>
  <LastModified>2014-11-21T19:40:05.000Z</LastModified>
  <ETag>&quot;70ee1738b6b21e2c8a43f3a5ab0eee71&quot;</ETag>
  <Size>11</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
...
</ListBucketResult>
```

In the following subsequent request, we include a `continuation-token` query parameter in the request with value of the `<NextContinuationToken>` from the preceding response.

```
GET /?list-type=2 HTTP/1.1
GET /?list-type=2&continuation-token=1ueGcxLPRx1Tr/XYEExHnhbYLgveDs2J/wm36Hy4vb0wM= HTTP/1.1

Host: bucket.s3.amazonaws.com
Date: Mon, 02 May 2016 23:17:07 GMT
Authorization: authorization string
```

Amazon S3 returns a list of the next set of keys starting where the previous request ended.

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>bucket</Name>
  <Prefix></Prefix>
  <ContinuationToken>1ueGcxLPRx1Tr/XYEExHnhbYLgveDs2J/wm36Hy4vb0wM=</ContinuationToken>
  <KeyCount>112</KeyCount>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>happyfacex.jpg</Key>
    <LastModified>2014-11-21T19:40:05.000Z</LastModified>
    <ETag>&quot;70ee1738b6b21e2c8a43f3a5ab0eee71&quot;</ETag>
    <Size>1111</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  ...
</ListBucketResult>
```

## More Info

- [GET Object \(p. 349\)](#)
- [PUT Object \(p. 412\)](#)
- [PUT Bucket \(p. 227\)](#)

# GET Bucket (List Objects) Version 1

## Description

### Important

This API has been revised. We recommend that you use the newer version, GET Bucket (List Objects) version 2, when developing applications. For more information, see [GET Bucket \(List Objects\) Version 2 \(p. 101\)](#). For backward compatibility, Amazon S3 continues to support GET Bucket (List Objects) version 1.

This implementation of the `GET` operation returns some or all (up to 1,000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. A `200 OK` response can contain valid or invalid XML. Be sure to design your application to parse the contents of the response and handle it appropriately.

To use this implementation of the operation, you must have `READ` access to the bucket.

### Note

To get a list of your buckets, see [GET Service \(p. 65\)](#).

## Requests

### Syntax

```
GET / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of `GET` uses the parameters in the following table to return a subset of the objects in a bucket.

Parameter	Description	Required
<code>delimiter</code>	A delimiter is a character you use to group keys.  If you specify a <code>prefix</code> , all keys that contain the same string between the <code>prefix</code> and the first occurrence of the delimiter after the prefix are grouped under a single result element called <code>CommonPrefixes</code> . If you don't specify the <code>prefix</code> parameter, the substring starts at the beginning of the key. The keys that are grouped under the <code>CommonPrefixes</code> result element are not returned elsewhere in the response.  Type: String  Default: None	No
<code>encodingtype</code>	Requests Amazon S3 to encode the response and specifies the encoding method to use.  An object key can contain any Unicode character. However, XML 1.0 parsers cannot parse some characters, such as characters with an ASCII value from	No

Parameter	Description	Required
	0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.  Type: String  Default: None  Valid value: url	
marker	Specifies the key to start with when listing objects in a bucket. Amazon S3 returns object keys in UTF-8 binary order, starting with key after the marker in order.  Type: String  Default: None	No
max-keys	Sets the maximum number of keys returned in the response body. If you want to retrieve fewer than the default 1,000 keys, you can add this to your request.  The response might contain fewer keys, but it never contains more. If there are additional keys that satisfy the search criteria, but these keys were not returned because max-keys was exceeded, the response contains <IsTruncated>true</IsTruncated>. To return the additional keys, see marker.  Type: String  Default: 1,000	No
prefix	Limits the response to keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using prefix to make groups in the same way you would use a folder in a file system.)  Type: String  Default: None	No

## Request Elements

This implementation of the operation does not use request elements.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

Name	Description
Contents	<p>Metadata about each object returned.</p> <p>Type: XML metadata</p> <p>Ancestor: <code>ListBucketResult</code></p>
CommonPrefixes	<p>All of the keys rolled up in a common prefix count as a single return when calculating the number of returns. See <code>MaxKeys</code>.</p> <ul style="list-style-type: none"> <li>A response can contain <code>CommonPrefixes</code> only if you specify a delimiter.</li> <li><code>CommonPrefixes</code> contains all (if there are any) keys between <code>Prefix</code> and the next occurrence of the string specified by the delimiter.</li> <li><code>CommonPrefixes</code> lists keys that act like subdirectories in the directory specified by <code>Prefix</code>.</li> </ul> <p>For example, if the prefix is <code>notes/</code> and the delimiter is a slash (/) as in <code>notes/summer/july</code>, the common prefix is <code>notes/summer/</code>. All of the keys that roll up into a common prefix count as a single return when calculating the number of returns. See <code>MaxKeys</code>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
Delimiter	<p>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the <code>CommonPrefixes</code> collection. These rolled-up keys are not returned elsewhere in the response. Each rolled-up result counts as only one return against the <code>MaxKeys</code> value.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
DisplayName	<p>Object owner's name.</p> <p><b>Important</b> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents.Owner</code></p>
Encoding-Type	Encoding type used by Amazon S3 to encode object key names in the XML response.

Name	Description
	<p>If you specify the <code>encodingtype</code> request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements:</p> <p><code>Delimiter</code>, <code>Marker</code>, <code>Prefix</code>, <code>NextMarker</code>, <code>Key</code></p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>ETag</code>	<p>The entity tag is an MD5 hash of the object. The <code>ETag</code> reflects only changes to the contents of an object, not its metadata.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>
<code>ID</code>	<p>Object owner's ID.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents.Owner</code></p>
<code>IsTruncated</code>	<p>Specifies whether (<code>true</code>) or not (<code>false</code>) all of the results were returned. If the number of results exceeds that specified by <code>MaxKeys</code>, all of the results might not be returned.</p> <p>Type: Boolean</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>Key</code>	<p>The object's key.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>
<code>LastModified</code>	<p>Date and time the object was last modified.</p> <p>Type: Date</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>
<code>Marker</code>	<p>Indicates where in the bucket listing begins. <code>Marker</code> is included in the response if it was sent with the request.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>MaxKeys</code>	<p>The maximum number of keys returned in the response body.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>

Name	Description
Name	<p>Name of the bucket.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
NextMarker	<p>When the response is truncated (that is, the <code>IsTruncated</code> element value in the response is true), you can use the key name in this field as a marker in the subsequent request to get next set of objects. Amazon S3 lists objects in UTF-8 character encoding in lexicographical order.</p> <p><b>Note</b></p> <p>This element is returned only if you specify a delimiter request parameter. If the response does not include the <code>NextMarker</code> and it is truncated, you can use the value of the last Key in the response as the marker in the subsequent request to get the next set of object keys.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
Owner	<p>Bucket owner.</p> <p>Type: String</p> <p>Children: <code>DisplayName</code>, <code>ID</code></p> <p>Ancestor: <code>ListBucketResult.Contents</code>   <code>CommonPrefixes</code></p>
Prefix	<p>Keys that begin with the indicated prefix.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
Size	<p>Size in bytes of the object.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>
StorageClass	<p><code>STANDARD</code>   <code>STANDARD_IA</code>   <code>ONEZONE_IA</code>   <code>REDUCED_REDUNDANCY</code>   <code>GLACIER</code></p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult.Contents</code></p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request returns the objects in BucketName.

```
GET / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>bucket</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>my-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>my-third-image.jpg</Key>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"1b2cf535f27731c974343645a3985328"</ETag>
    <Size>64994</Size>
    <StorageClass>STANDARD_IA</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

### Sample Request Using Request Parameters

This example lists up to 40 keys in the quotes bucket that start with N and occur lexicographically after Ned.

```
GET /?prefix=N&marker=Ned&max-keys=40 HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
```

```
x-amz-request-id: 3B3C7C725673C630
Date: Wed, 01 Mar 2006 12:00:00 GMT
Content-Type: application/xml
Content-Length: 302
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>quotes</Name>
  <Prefix>N</Prefix>
  <Marker>Ned</Marker>
  <MaxKeys>40</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>Nelson</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fd9f00ad9f27c383fc9ac7f"</ETag>
    <Size>5</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaf161ca5fb16fd081034f</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>Neo</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fd9f00ad9f27c383fc9ac7f"</ETag>
    <Size>4</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaf1ffd86a5fb16fd081034f</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

## Sample Request Using a Prefix and Delimiter

For this example, we assume that you have the following keys in your bucket:

```
sample.jpg
photos/2006/January/sample.jpg
photos/2006/February/sample2.jpg
photos/2006/February/sample3.jpg
photos/2006/February/sample4.jpg
```

The following GET request specifies the `delimiter` parameter with value `/`.

```
GET /?delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

The key `sample.jpg` does not contain the delimiter character, and Amazon S3 returns it in the `Contents` element in the response. However, all other keys contain the delimiter character. Amazon S3 groups these keys and returns a single `CommonPrefixes` element with prefix value `photos/` that is a substring from the beginning of these keys to the first occurrence of the specified delimiter.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>example-bucket</Name>
<Prefix></Prefix>
<Marker></Marker>
<MaxKeys>1000</MaxKeys>
<Delimiter>/</Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
  <Key>sample.jpg</Key>
  <LastModified>2011-02-26T01:56:20.000Z</LastModified>
  <ETag>"bf1d737a4d46a19f3bcd6905cc8b902"</ETag>
  <Size>142863</Size>
  <Owner>
    <ID>canonical-user-id</ID>
    <DisplayName>display-name</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<CommonPrefixes>
  <Prefix>photos/</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

The following GET request specifies the `delimiter` parameter with the value `/`, and the `prefix` parameter with the value `photos/2006/`.

```
GET /?prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

In response, Amazon S3 returns only the keys that start with the specified prefix. It uses the `delimiter` character to group keys that contain the same substring until the first occurrence of the `delimiter` character after the specified prefix. For each such key group, Amazon S3 returns one `<CommonPrefixes>` element in the response. The keys grouped under this `CommonPrefixes` element are not returned elsewhere in the response. The value returned in the `CommonPrefixes` element is a substring that starts at the beginning of the key and ends at the first occurrence of the specified delimiter after the prefix.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>example-bucket</Name>
<Prefix>photos/2006/</Prefix>
<Marker></Marker>
<MaxKeys>1000</MaxKeys>
<Delimiter>/</Delimiter>
<IsTruncated>false</IsTruncated>

<CommonPrefixes>
  <Prefix>photos/2006/February/</Prefix>
</CommonPrefixes>
<CommonPrefixes>
  <Prefix>photos/2006/January/</Prefix>
</CommonPrefixes>
</ListBucketResult>
```

## Related Resources

- [GET Object \(p. 349\)](#)
- [PUT Object \(p. 412\)](#)

- [PUT Bucket \(p. 227\)](#)

# GET Bucket accelerate

## Description

This implementation of the `GET` operation uses the `accelerate` subresource to return the Transfer Acceleration state of a bucket, which is either `Enabled` or `Suspended`. Amazon S3 Transfer Acceleration is a bucket-level feature that enables you to perform faster data transfers to and from Amazon S3.

To use this operation, you must have permission to perform the `s3:GetAccelerateConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

You set the Transfer Acceleration state of an existing bucket to `Enabled` or `Suspended` by using the [PUT Bucket accelerate \(p. 232\)](#) operation.

A GET `accelerate` request does not return a state value for a bucket that has no transfer acceleration state. A bucket has no Transfer Acceleration state, if a state has never been set on the bucket.

This implementation of the `GET` operation returns the following responses:

- If the transfer acceleration state is set to `Enabled` on a bucket, the response is:

```
<AccelerateConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</AccelerateConfiguration>
```

- If the transfer acceleration state is set to `Suspended` on a bucket, the response is:

```
<AccelerateConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Suspended</Status>
</AccelerateConfiguration>
```

- If the transfer acceleration state on a bucket has never been set to `Enabled` or `Suspended`, the response is:

```
<AccelerateConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/" />
```

For more information on transfer acceleration, see [Transfer Acceleration in the Amazon Simple Storage Service Developer Guide](#).

## Requests

### Syntax

```
GET /?accelerate HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of `GET` returns the following response elements.

Name	Description
AccelerateConfiguration	Container for the Status response element.  Type: Container  Ancestor: None
Status	The transfer acceleration state of the bucket.  Type: Enum  Valid Values: Suspended   Enabled  Ancestor: AccelerateConfiguration

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Retrieve the transfer acceleration configuration for a bucket

The following example shows a `GET /?accelerate` request to retrieve the transfer acceleration state of the bucket named `examplebucket`.

```
GET /?accelerate HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

The following is a sample of the response body (only) that shows bucket transfer acceleration is enabled.

```
<AccelerateConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Status>Enabled</Status>
</AccelerateConfiguration>
```

## Related Resources

- [PUT Bucket accelerate \(p. 232\)](#)

# GET Bucket acl

## Description

This implementation of the `GET` operation uses the `acl` subresource to return the access control list (ACL) of a bucket. To use `GET` to return the ACL of the bucket, you must have `READ_ACP` access to the bucket. If `READ_ACP` permission is granted to the anonymous user, you can return the ACL of the bucket without using an authorization header.

## Requests

### Syntax

```
GET /?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
AccessControlList	Container for ACL information.  Type: Container  Ancestry: AccessControlPolicy
AccessControlPolicy	Container for the response.

Name	Description
	Type: Container  Ancestry: None
DisplayName	Bucket owner's display name. This is returned only if the owner's e-mail address (or the forum name, if configured) can be determined from the ID.  <b>Important</b> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .  Type: String  Ancestry: AccessControlPolicy.Owner
Grant	Container for Grantee and Permission.  Type: Container  Ancestry: AccessControlPolicy.AccessControlList
Grantee	Container for DisplayName and ID of the person being granted permissions.  Type: Container  Ancestry: AccessControlPolicy.AccessControlList.Grant
ID	Bucket owner's ID.  Type: String  Ancestry: AccessControlPolicy.Owner
Owner	Container for bucket owner information.  Type: Container  Ancestry: AccessControlPolicy
Permission	Permission given to the Grantee for bucket.  Type: String  Valid Values: FULL_CONTROL   WRITE   WRITE_ACP   READ   READ_ACP  Ancestry: AccessControlPolicy.AccessControlList.Grant

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the ACL of the specified bucket.

```
GET ?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
        <DisplayName>CustomersName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

## Related Resources

- [GET Bucket Objects \(p. 111\)](#)

# GET Bucket analytics

## Description

This implementation of the GET operation returns an analytics configuration (identified by the analytics configuration ID) from the bucket.

To use this operation, you must have permissions to perform the `s3:GetAnalyticsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about Amazon S3 analytics feature, see [Amazon S3 Analytics – Storage Class Analysis](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?analytics&id=analytics-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of GET uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID that identifies the analytics configuration. Limited to 64 characters.  Type: String  Default: None  Valid Characters for <code>id</code> : a-z A-Z 0-9 - _ .	Yes

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

# Responses

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

The Examples section shows an example of an analytics configuration XML. The following table describes the XML elements in the analytics configuration returned by the GET request.

Name	Description
AnalyticsConfiguration	Contains the configuration and any analyses for the analytics filter.  Type: Container  Children: <code>Id</code> , <code>Filter</code> , <code>StorageClassAnalysis</code>  Ancestor: None
And	A conjunction (logical AND) of predicates, which is used in evaluating an analytics filter. The operator must have at least two predicates.  Type: String  Children: <code>Prefix</code> , <code>Tag</code>  Ancestor: <code>Filter</code>
Bucket	The Amazon Resource Name (ARN) of the bucket where analytics results are published.  Type: String  Ancestor: <code>S3BucketDestination</code>
BucketAccountId	The ID of the account that owns the destination bucket where the analytics results are published.  Type: String  Ancestor: <code>S3BucketDestination</code>
DataExport	A container used to describe how data related to the storage class analysis should be exported.  Type: Container  Children: <code>OutputSchemaVersion</code> , <code>Destination</code>  Ancestor: <code>StorageClassAnalysis</code>
Destination	Contains information about where to publish the analytics results.  Type: Container  Children: <code>S3BucketDestination</code>

Name	Description
	Ancestor: <code>DataExport</code>
<code>Filter</code>	<p>Specifies an analytics filter. The analytics only includes objects that meet the filter's criteria.</p> <p>Type: Container</p> <p>Children: And</p> <p>Ancestor: <code>AnalyticsConfiguration</code></p>
<code>Format</code>	<p>Specifies the output format of the analytics results. Currently, Amazon S3 supports the comma-separated value (CSV) format.</p> <p>Type: String</p> <p>Ancestor: <code>S3BucketDestination</code></p> <p>Valid values: CSV</p>
<code>Id</code>	<p>The ID that identifies the analytics configuration.</p> <p>Type: String</p> <p>Ancestor: <code>AnalyticsConfiguration</code></p>
<code>Key</code>	<p>The key for a tag.</p> <p>Type: String</p> <p>Ancestor: Tag</p>
<code>OutputSchemaVersion</code>	<p>The version of the output schema to use when exporting data. Must be V_1.</p> <p>Type: String</p> <p>Ancestor: <code>DataExport</code></p> <p>Valid values: v_1</p>
<code>Prefix</code>	<p>The prefix that an object must have to be included in the analytics results.</p> <p>Type: String</p> <p>Ancestor: And</p>
<code>Prefix</code>	<p>The prefix that is prepended to all analytics results.</p> <p>Type: String</p> <p>Ancestor: <code>S3BucketDestination</code></p>

Name	Description
StorageClassAnalysis	If present, it indicates that data related to access patterns is collected and made available to analyze the tradeoffs between different storage classes.  Type: Container  Children: DataExport  Ancestor: AnalyticsConfiguration
S3BucketDestination	Contains the bucket ARN, file format, bucket owner (optional), and prefix (optional) where analytics results are published.  Type: Container  Children: Format, BucketAccountId, Bucket, Prefix  Ancestor: Destination.
Tag	The tag to use when evaluating an analytics filter.  Type: Container  Children: Key, Value  Ancestor: And
Value	The value for a tag.  Type: String  Ancestor: Tag

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example: Configure an Analytics Report

The following GET request for the bucket examplebucket returns the inventory configuration with the ID list1.

```
GET /?analytics&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
```

```
Date: Mon, 31 Oct 2016 12:00:00 GMT
Server: AmazonS3
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>list1</Id>
  <Filter>
    <And>
      <Prefix>images/</Prefix>
      <Tag>
        <Key>dog</Key>
        <Value>corgi</Value>
      </Tag>
    </And>
  </Filter>
  <StorageClassAnalysis>
    <DataExport>
      <OutputSchemaVersion>V_1</OutputSchemaVersion>
      <Destination>
        <S3BucketDestination>
          <Format>CSV</Format>
          <BucketAccountId>123456789012</BucketAccountId>
          <Bucket>arn:aws:s3:::destination-bucket</Bucket>
          <Prefix>destination-prefix</Prefix>
        </S3BucketDestination>
      </Destination>
    </DataExport>
  </StorageClassAnalysis>
</AnalyticsConfiguration>
```

## Related Resources

- [DELETE Bucket analytics \(p. 80\)](#)
- [List Bucket Analytics Configurations \(p. 206\)](#)
- [PUT Bucket analytics \(p. 242\)](#)

# GET Bucket cors

## Description

Returns the `cors` configuration information set for the bucket.

To use this operation, you must have permission to perform the `s3:GetBucketCORS` action. By default, the bucket owner has this permission and can grant it to others.

To learn more `cors`, go to [Enabling Cross-Origin Resource Sharing](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?cors HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of GET returns the following response elements.

Name	Description
CORSConfiguration	Container for up to 100 CORSRules elements.  Type: Container  Children: CORSRules

Name	Description
	Ancestor: None
CORSRule	<p>A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.</p> <p>Type: Container</p> <p>Children: AllowedOrigin, AllowedMethod, MaxAgeSeconds, ExposeHeader, ID.</p> <p>Ancestor: CORSConfiguration</p>
AllowedHeader	<p>Specifies which headers are allowed in a pre-flight OPTIONS request through the Access-Control-Request-Headers header. Each header name specified in the Access-Control-Request-Headers must have a corresponding entry in the rule. Only the headers that were requested will be sent back. This element can contain at most one * wildcard character.</p> <p>A CORSRule can have at most one MaxAgeSeconds element.</p> <p>Type: Integer (seconds)</p> <p>Ancestor: CORSRule</p>
AllowedMethod	<p>Identifies an HTTP method that the domain/origin specified in the rule is allowed to execute.</p> <p>Each CORSRule must contain at least one AllowedMethod and one AllowedOrigin element.</p> <p>Type: Enum (GET, PUT, HEAD, POST, DELETE)</p> <p>Ancestor: CORSRule</p>
AllowedOrigin	<p>One or more response headers that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).</p> <p>Each CORSRule must have at least one AllowedOrigin element. The string value can include at most one '*' wildcard character, for example, http://*.example.com". You can also specify only "*" to allow cross-origin access for all domains/origins.</p> <p>Type: String</p> <p>Ancestor: CORSRule</p>
ExposeHeader	<p>One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).</p> <p>You add one ExposeHeader in the rule for each header.</p> <p>Type: String</p> <p>Ancestor: CORSRule</p>

Name	Description
ID	An optional unique identifier for the rule. The ID value can be up to 255 characters long. The IDs help you find a rule in the configuration.  Type: String  Ancestor: CORSRule
MaxAgeSeconds	The time in seconds that your browser is to cache the preflight response for the specified resource.  A CORSRule can have at most one MaxAgeSeconds element.  Type: Integer (seconds)  Ancestor: CORSRule

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Retrieve cors subresource

The following example gets the cors subresource of a bucket.

#### Sample Request

```
GET /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 13 Dec 2011 19:14:42 GMT
Authorization: signatureValue
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: OFmFIWsh/PpBuzzOJFRC55ZGVmQW4SHJ7xVDqKwhEdJmf3q63RtrvH8ZuxW1B0l5
x-amz-request-id: OCF038E9BCF63097
Date: Tue, 13 Dec 2011 19:14:42 GMT
Server: AmazonS3
Content-Length: 280

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSec>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
</CORSConfiguration>
```

## Related Resources

- [PUT Bucket cors \(p. 248\)](#)
- [DELETE Bucket cors \(p. 82\)](#)
- [OPTIONS object \(p. 382\)](#)

# GET Bucket encryption

## Description

This implementation of the GET operation uses the encryption subresource to return the default encryption configuration for an Amazon S3 bucket. For information about the Amazon S3 default encryption feature, see [Amazon S3 Default Bucket Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

To use this operation, you must have permission to perform the s3:GetEncryptionConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?encryption HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of GET returns the following response elements.

Name	Description	
ApplyServerSideEncryptionByDefault	Container for setting server-	

Name	Description	
	<p>side encryption by default.</p> <p>Type: Container</p> <p>Children: SSEAlgorithm, KMSMasterKeyID</p> <p>Ancestor: Rule</p>	
KMSMasterKeyID	<p>The AWS KMS master key ID used for the SSE-KMS encryption.</p> <p>Type: String</p> <p>Ancestor: ApplyServerSideEncryptionByDefault</p> <p>Constraint: Can only be used when you set the value of SSEAlgorithm as aws:kms. The default aws:s3 AWS KMS master key is used if this element is absent while the SSEAlgorithm is aws:kms.</p>	
Rule	<p>Container for server-side encryption by default configuration.</p> <p>Type: Container</p> <p>Children: ApplyServerSideEncryptionByDefault</p> <p>Ancestor: ServerSideEncryptionConfiguration</p>	
ServerSideEncryptionConfiguration	<p>Container for the server-side encryption by default configuration rule.</p> <p>Type: Container</p> <p>Children: Rule</p> <p>Ancestor: None</p>	

Name	Description
SSEAlgorithm	<p>The server-side encryption algorithm to use.</p> <p>Type: String</p> <p>Valid Values: AES256, aws:kms</p> <p>Ancestor: ApplyServerSideEncryptionByDefault</p> <p>Constraint: Can only be used when you use ApplyServerSideEncryptionByDefault.</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Retrieve the Encryption Configuration for an S3 Bucket

The following example shows a GET /?encryption request.

```
GET /?encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: authorization string
Content-Length: length
```

The following is a sample of the response.

```
HTTP/1.1 200 OK
x-amz-id-2: kDmqsuw5FDmgLmxQaUkd9A4NJ/PIiE0c1rAU/ue2Yp60toXs4I5k5fqlwZsA6fV+wJQCzRRwygQ=
x-amz-request-id: 5D8706FCB2673B7D
Date: Wed, 06 Sep 2017 12:00:00 GMT
Transfer-Encoding: chunked
Server: AmazonS3

<ServerSideEncryptionConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      < SSEAlgorithm>aws:kms</SSEAlgorithm>
      < KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>
```

## Related Resources

- [PUT Bucket encryption \(p. 254\)](#)
- [DELETE Bucket encryption \(p. 84\)](#)

# GET Bucket Inventory

## Description

This implementation of the `GET` operation returns an inventory configuration (identified by the inventory configuration ID) from the bucket.

To use this operation, you must have permissions to perform the `s3:GetInventoryConfiguration` action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about the Amazon S3 inventory feature, see [Amazon S3 Inventory](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of `GET` uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID that identifies the inventory configuration.  Type: String  Default: None  Valid Characters for <code>id</code> : a-z A-Z 0-9 - _ .	Yes

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

# Responses

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

The Examples section shows an example of an inventory configuration XML. The following table describes the XML elements in the inventory configuration returned by the GET request.

Name	Description
AccountId	<p>The ID of the account that owns the destination bucket where the inventory is published.</p> <p>Although optional, we recommend that the value be set to prevent problems if the destination bucket ownership changes.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p>
Bucket	<p>The Amazon Resource Name (ARN) of the bucket where inventory results are published.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p>
Destination	<p>Contains information about where to publish the inventory results.</p> <p>Type: Container</p> <p>Children: S3BucketDestination</p> <p>Ancestor: InventoryConfiguration</p>
Encryption	<p>Contains the type of server-side encryption used to encrypt the inventory results.</p> <p>Type: Container</p> <p>Children: SSE-KMS, SSE-S3</p> <p>Ancestor: S3BucketDestination</p>
Field	<p>Contains the optional fields that are included in the inventory results. Multiple Field elements can be contained in OptionalFields.</p> <p>Type: String</p> <p>Ancestor: OptionalFields</p> <p>Valid values: Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode, ObjectLockLegalHoldStatus</p>

Name	Description
<code>Filter</code>	<p>Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria.</p> <p>Type: Container</p> <p>Children: <code>Prefix</code></p> <p>Ancestor: <code>InventoryConfiguration</code></p>
<code>Format</code>	<p>Specifies the output format of the inventory results. Currently, Amazon S3 supports the comma-separated values (CSV) format, the <a href="#">Apache optimized row columnar (ORC)</a> format, and the <a href="#">Apache Parquet (Parquet)</a> format.</p> <p>Type: String</p> <p>Ancestor: <code>S3BucketDestination</code></p> <p>Valid values: CSV, ORC, or Parquet</p>
<code>Frequency</code>	<p>Specifies how frequently inventory results are produced.</p> <p>Type: String</p> <p>Ancestor: <code>Schedule</code></p> <p>Valid values: Daily, or Weekly</p>
<code>Id</code>	<p>The ID that identifies the inventory configuration.</p> <p>Type: String</p> <p>Ancestor: <code>InventoryConfiguration</code></p>
<code>IncludedObjectVersions</code>	<p>Object versions to include in the inventory list. If set to All, the list includes all the object versions, which adds the version-related fields <code>VersionId</code>, <code>IsLatest</code>, and <code>DeleteMarker</code> to the list. If set to Current, the list does not contain these version-related fields.</p> <p>Type: String</p> <p>Ancestor: <code>InventoryConfiguration</code></p> <p>Valid values: Current or All</p>
<code>InventoryConfiguration</code>	<p>Contains the inventory configuration.</p> <p>Type: Container</p> <p>Children: <code>Id</code>, <code>.IsEnabled</code>, <code>Filter</code>, <code>Destination</code>, <code>Schedule</code>, <code>IncludedObjectVersions</code>, and <code>OptionalFields</code> elements.</p> <p>Ancestor: None</p>

Name	Description
Enabled	<p>Specifies whether the inventory is enabled or disabled. If set to <code>True</code>, an inventory list is generated. If set to <code>False</code>, no inventory list is generated.</p> <p>Type: String</p> <p>Ancestor: <code>InventoryConfiguration</code></p> <p>Valid values: <code>True</code> or <code>False</code></p>
KeyId	<p>The AWS KMS customer master key (CMK) used to encrypt the inventory file.</p> <p>Type: String</p> <p>Ancestor: <code>SSE-KMS</code></p> <p>Valid values: ARN of the CMK</p>
OptionalFields	<p>Contains the optional fields.</p> <p>Type: Container</p> <p>Children: <code>Field</code></p> <p>Ancestor: <code>InventoryConfiguration</code></p>
Prefix	<p>The prefix that an object must have to be included in the inventory results.</p> <p>Type: String</p> <p>Ancestor: <code>Filter</code></p>
Prefix	<p>The prefix that is prepended to all inventory results.</p> <p>Type: String</p> <p>Ancestor: <code>S3BucketDestination</code></p>
Schedule	<p>Contains the frequency of inventory results generation.</p> <p>Type: Container</p> <p>Children: <code>Frequency</code></p> <p>Ancestor: <code>Destination</code>.</p>
SSE-KMS	<p>Specifies to use server-side encryption with AWS KMS-managed keys (SSE-KMS) and contains the key that is used to encrypt the inventory file.</p> <p>Type: Container</p> <p>Children: <code>KeyId</code></p> <p>Ancestor: <code>Encryption</code></p>

Name	Description
SSE-S3	<p>Specifies to use server-side encryption with Amazon S3-managed keys (SSE-S3) to encrypt the inventory file.</p> <p>Type: Container</p> <p>Ancestor: Encryption</p> <p>Valid values: empty</p>
S3BucketDestination	<p>Contains the bucket ARN, file format, bucket owner (optional), prefix where inventory results are published (optional), and the type of server-side encryption that is used to encrypt the file (optional).</p> <p>Type: Container</p> <p>Children: Format, AccountId, Bucket, Prefix, Encryption</p> <p>Ancestor: Destination.</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example: Configure an Inventory Report

The following GET request for the bucket examplebucket returns the inventory configuration with the ID list1.

```
GET /?inventory&id=list1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bjOKMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Mon, 31 Oct 2016 12:00:00 GMT
Server: AmazonS3
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <isEnabled>true</isEnabled>
  <Destination>
    <S3BucketDestination>
      <Format>CSV</Format>
      <AccountId>123456789012</AccountId>
      <Bucket>arn:aws:s3:::destination-bucket</Bucket>
      <Prefix>prefix1</Prefix>
      <SSE-S3/>
    </S3BucketDestination>
  </Destination>
</InventoryConfiguration>
```

```
</S3BucketDestination>
</Destination>
<Schedule>
  <Frequency>Daily</Frequency>
</Schedule>
<Filter>
  <Prefix>myprefix/<Prefix>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
<OptionalFields>
  <Field>Size</Field>
  <Field>LastModifiedDate</Field>
  <Field>ETag</Field>
  <Field>StorageClass</Field>
  <Field>IsMultipartUploaded</Field>
  <Field>ReplicationStatus</Field>
  <Field>ObjectLockRetainUntilDate</Field>
  <Field>ObjectLockMode</Field>
  <Field>ObjectLockLegalHoldStatus</Field>
</OptionalFields>
</InventoryConfiguration>
```

## Related Resources

- [DELETE Bucket inventory \(p. 86\)](#)
- [List Bucket Inventory Configurations \(p. 210\)](#)
- [PUT Bucket inventory \(p. 258\)](#)

# GET Bucket lifecycle

## Description

### Note

Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both. Accordingly, this section describes the latest API. The response describes the new filter element that you can use to specify a filter to select a subset of objects to which the rule applies. If you are still using previous version of the lifecycle configuration, it works. For related API description, see [GET Bucket lifecycle \(Deprecated\) \(p. 579\)](#).

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, go to [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

To use this operation, you must have permission to perform the `s3:GetLifecycleConfiguration` action. The bucket owner has this permission, by default. The bucket owner can grant this permission to others. For more information about permissions, see [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of GET returns the following response elements.

Name	Description
And	<p>Container for specifying <code>Prefix</code> and <code>Tag</code> based filters.</p> <p>Child: <code>Prefix</code> and <code>Tag</code></p> <p>Type: Container</p> <p>Ancestor: <code>Filter</code></p>
AbortIncompleteMultipartUpload	<p>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.</p> <p>Child: <code>DaysAfterInitiation</code></p> <p>Type: Container</p> <p>Ancestor: <code>Rule</code></p>
Date	<p>Date when you want Amazon S3 to take the action. For more information, see <a href="#">Lifecycle Rules: Based on a Specific Date</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Ancestor: <code>Expiration</code> or <code>Transition</code></p>
Days	<p>Specifies the number of days after object creation when the specific rule action takes effect. The object's eligibility time is calculated as creation time + the number of days, and rounding the resulting time to the next day midnight UTC.</p> <p>Type: Non-negative Integer when used with <code>Transition</code>, Positive Integer when used with <code>Expiration</code></p> <p>Ancestor: <code>Transition</code> or <code>Expiration</code></p>
DaysAfterInitiation	<p>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload.</p> <p>Type: Positive Integer</p> <p>Ancestor: <code>AbortIncompleteMultipartUpload</code></p>
Expiration	<p>This action specifies a period in the object's lifetime when Amazon S3 should take the appropriate expiration action. The expiration action occurs only on objects that are eligible according to the period specified in the child <code>Date</code> or <code>Days</code> element. The action Amazon S3 takes depends on whether the bucket is versioning enabled.</p> <ul style="list-style-type: none"> <li>• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</li> </ul>

Name	Description
	<ul style="list-style-type: none"> <li>Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. Buckets with versioning-enabled or versioning-suspended can have many versions of the same object, one current version, and zero or more noncurrent versions.</li> </ul> <p>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</p> <p><b>Important</b> If the state of your bucket is versioning-suspended, Amazon S3 creates a delete marker with version ID <code>null</code>. If you have a version with version ID <code>null</code>, Amazon S3 overwrites that version.</p> <p><b>Note</b> To set the expiration for noncurrent objects, you must use the <code>NoncurrentVersionExpiration</code> action.</p> <p>Type: Container Children: Days or Date Ancestor: Rule</p>
<code>Filter</code>	<p>Container element describing one or more filters used to identify a subset of objects to which the lifecycle rule applies.</p> <p>Child: <code>Prefix</code>, <code>Tag</code>, or <code>And</code> (if both prefix and tag are specified)</p> <p>Type: String Ancestor: Rule</p>
<code>ID</code>	<p>Unique identifier for the rule. The value cannot be longer than 255 characters.</p> <p>Type: String Ancestor: Rule</p>
<code>Key</code>	<p>Tag key.</p> <p>Type: String Ancestor: Tag</p>

Name	Description
LifecycleConfiguration	<p>Container for lifecycle rules. You can add as many as 1000 rules.</p> <p>Type: Container</p> <p>Children: Rule</p> <p>Ancestor: None</p>
ExpiredObjectDeleteMarker	<p>On a versioned bucket (a versioning-enabled or versioning-suspended bucket), this element indicates whether Amazon S3 will delete any expired object delete markers in the bucket. For an example, go to <a href="#">Example 8: Specify Expiration Action to Remove Expired Object Delete Markers</a> in the Amazon Simple Storage Service Developer Guide.</p> <p>Type: String</p> <p>Valid values: true   false (the value false is allowed, but it is no-op, which means that Amazon S3 does not take action if the value is false)</p> <p>Ancestor: Expiration</p>
NoncurrentDays	<p>Specifies the number of days that an object is noncurrent before Amazon S3 can perform the associated action. For information about calculating noncurrent days, see <a href="#">Lifecycle Rules Based on the Number of Days</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration</p> <p>Ancestor: NoncurrentVersionExpiration or NoncurrentVersionTransition</p>
NoncurrentVersionExpiration	<p>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.</p> <p>Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: NoncurrentDays</p> <p>Ancestor: Rule</p>

Name	Description
NoncurrentVersionTransition	<p>Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class.</p> <p>If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request Amazon S3 to transition noncurrent object versions to the GLACIER storage class at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: NoncurrentDays and StorageClass</p> <p>Ancestor: Rule</p>
Prefix	<p>Object key prefix identifying one or more objects to which the rule applies.</p> <p>Type: String</p> <p>Ancestor: Filter or And (if you specify Prefix and Tag child elements in the Filter)</p>
Rule	<p>Container for a lifecycle rule.</p> <p>Type: Container</p> <p>Ancestor: LifecycleConfiguration</p>
Status	<p>If enabled, Amazon S3 executes the rule as scheduled. If it is disabled, Amazon S3 ignores the rule.</p> <p>Type: String</p> <p>Ancestor: Rule</p> <p>Valid values: Enabled or Disabled</p>
StorageClass	<p>Specifies the Amazon S3 storage class to which you want to transition the object.</p> <p>Type: String</p> <p>Ancestor: Transition and NoncurrentVersionTransition</p> <p>Valid values: STANDARD_IA   ONEZONE_IA   GLACIER</p>
Tag	<p>Container listing the tag key and value used to filter objects to which the rule applies.</p> <p>Type: String</p> <p>Ancestor: Filter</p>

Name	Description
Transition	<p>This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.</p> <ul style="list-style-type: none"> <li>If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.</li> <li>If your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.</li> </ul> <p><b>Note</b> A versioning-enabled or versioning-suspended bucket can contain many versions of an object. This action has no effect on noncurrent object versions. To transition noncurrent objects, you must use the <code>NoncurrentVersionTransition</code> action.</p> <p>Type: Container Children: Days or Date, and StorageClass Ancestor: Rule</p>
Value	<p>Tag key value.</p> <p>Type: String Ancestor: Tag</p>

## Special Errors

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
<code>NoSuchLifecycleConfiguration</code>	The lifecycle configuration does not exist.	404 Not Found	Client

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Retrieve the Lifecycle Subresource

This example is a GET request to retrieve the `lifecycle` subresource from the specified bucket. The example response returns the lifecycle configuration.

## Sample Request

```
GET /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2012 00:17:21 GMT
Authorization: signatureValue
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hk1TXouTf0hccUjo0iCPjz6FnfIutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991C342C575321
Date: Thu, 15 Nov 2012 00:17:23 GMT
Server: AmazonS3
Content-Length: 358

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ID>Archive and then delete rule</ID>
    <Filter>
      <Prefix>projectdocs/<Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

## Related Resources

- [PUT Bucket lifecycle \(p. 265\)](#)
- [DELETE Bucket lifecycle \(p. 88\)](#)

# GET Bucket location

## Description

This implementation of the `GET` operation uses the `location` subresource to return a bucket's region. You set the bucket's region using the `LocationConstraint` request parameter in a `PUT Bucket` request. For more information, see [PUT Bucket \(p. 227\)](#).

To use this implementation of the operation, you must be the bucket owner.

## Requests

### Syntax

```
GET /?location HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
<code>LocationConstraint</code>	Specifies the region where the bucket resides.  Type: String  Valid Values: For a list of all the Amazon S3 supported location constraints by region, see <a href="#">Regions and Endpoints</a> in the <a href="#">AWS General Reference</a> .

Name	Description
	Ancestry: None

When the bucket's region is US East (N. Virginia), Amazon S3 returns an empty string for the bucket's region:

```
<LocationConstraint xmlns="http://s3.amazonaws.com/doc/2006-03-01/" />
```

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the region of the specified bucket.

```
GET /?location HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Tue, 09 Oct 2007 20:26:04 +0000
Authorization: signatureValue
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint xmlns="http://s3.amazonaws.com/doc/2006-03-01/">EU</LocationConstraint>
```

## Related Resources

- [GET Bucket Objects \(p. 111\)](#)
- [PUT Bucket \(p. 227\)](#)

# GET PublicAccessBlock

## Description

This operation retrieves the `PublicAccessBlock` configuration for an Amazon S3 bucket. In order to use this operation, you must have the `s3:GetBucketPublicAccessBlock` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

### Important

When Amazon S3 evaluates the `PublicAccessBlock` configuration for a bucket or an object, it checks the `PublicAccessBlock` configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the `PublicAccessBlock` settings are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see [The Meaning of "Public"](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
PublicAccessBlock	<p>AmazonS3Block configuration.</p> <p>Type: Container</p> <p>Children: BlockPublicAccls, IgnorePublicAccls, BlockPublicPolicy, RestrictPublicBuckets</p> <p>Ancestor: None</p>
BlockPublicAccls	<p>Specifies whether Amazon S3 will block public access control lists (ACLs) for this bucket and objects in this bucket.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>

Name	Description
IgnorePublicAcls	<p>Specifies whether Amazon S3 will ignore public ACLs for this bucket and objects in this bucket.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>
BlockPublicPolicy	<p>Specifies whether Amazon S3 will block public bucket policies for this bucket.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>
RestrictPublicBuckets	<p>Specifies whether Amazon S3 will restrict public bucket policies for this bucket.</p> <p>Type: Boolean</p> <p>Ancestor: PublicAccessBlockConfiguration</p> <p>Valid values: TRUE   FALSE</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request gets a bucket PublicAccessBlock configuration.

```
GET /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PublicAccessBlockConfiguration>
  <BlockPublicAcls>TRUE</BlockPublicAcls>
  <IgnorePublicAcls>FALSE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
```

```
</PublicAccessBlockConfiguration>
```

## Related Resources

- [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service Developer Guide*.
- [PUT PublicAccessBlock \(p. 277\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [GET PublicAccessBlock \(p. 69\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)
- [DELETE PublicAccessBlock \(p. 68\)](#)

# GET Bucket logging

## Note

Logging functionality is currently in beta.

## Description

This implementation of the GET operation uses the logging subresource to return the logging status of a bucket and the permissions users have to view and modify that status. To use GET, you must be the bucket owner.

## Requests

### Syntax

```
GET /?logging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
BucketLoggingStatus	Container for the response.  Type: Container  Ancestry: None
EmailAddress	E-mail address of the person whose logging permissions are displayed.

Name	Description
	Type: String  Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant.Grantee
Grant	Container for Grantee and Permission.  Type: Container  Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants
Grantee	Container for EmailAddress of the person whose logging permissions are displayed.  Type: Container  Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant
LoggingEnabled	Container for logging information. This element and its children are present when logging is enabled, otherwise, this element and its children are absent.  Type: Container  Ancestry: BucketLoggingStatus
Permission	Logging permissions assigned to the Grantee for the bucket.  Type: String  Valid Values: FULL_CONTROL   READ   WRITE  Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant
TargetBucket	Specifies the bucket whose logging status is being returned. This element specifies the bucket where server access logs will be delivered.  Type: String  Ancestry: BucketLoggingStatus.LoggingEnabled
TargetGrants	Container for granting information.  Type: Container  Ancestry: BucketLoggingStatus.LoggingEnabled
TargetPrefix	Specifies the prefix for the keys that the log files are being stored under.  Type: String  Ancestry: BucketLoggingStatus.LoggingEnabled

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the logging status for mybucket.

```
GET ?logging HTTP/1.1
Host: mybucket.s3.amazonaws.com
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string
```

### Sample Response Showing an Enabled Logging Status

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access_log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress>
        </Grantee>
        <Permission>READ</Permission>
      </Grant>
    </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>
```

### Sample Response Showing a Disabled Logging Status

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01" />
```

## Related Resources

- [PUT Bucket \(p. 227\)](#)
- [PUT Bucket logging \(p. 281\)](#)

# GET Bucket metrics

## Description

Gets a metrics configuration for the CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. Note that this doesn't include the daily storage metrics.

To use this operation, you must have permissions to perform the `s3:GetMetricsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about CloudWatch request metrics for Amazon S3, see [Monitoring Metrics with Amazon CloudWatch](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?metrics&id=id HTTP/1.1
Host: BucketName.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

Parameter	Description	Required
<code>id</code>	The ID used to identify the metrics configuration.	Yes

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

The Examples section shows an example of a metrics configuration XML. The following table describes the XML elements in the metrics configuration returned by the GET request.

Name	Description
And	A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates, and an object must match all of the predicates in order for the filter to apply.  Type: Container  Children: Prefix, Tag  Ancestor: Filter
Filter	Specifies a metrics configuration filter. The metrics configuration only includes objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (MetricsAndOperator).  Type: Container  Children: And  Ancestor: MetricsConfiguration
Id	The ID used to identify the metrics configuration.  Type: String  Ancestor: MetricsConfiguration
Key	The name of the tag.  Type: String  Ancestor: Tag
MetricsConfiguration	An existing metrics configuration for CloudWatch request metrics on this bucket.  Type: Container  Children: Filter, Id  Ancestor: None
Prefix	A string of text used at the beginning of an object key name.  Type: String  Ancestor: And
Tag	A key value name pair, used to organize objects by association.  Type: Container  Children: Key, Value  Ancestor: And
Value	The value of the tag.  Type: String

Name	Description
	Ancestor: Tag

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### First Sample Request

Retrieve a metrics configuration that filters metrics based on a specified prefix.

```
GET /?metrics&id=Documents HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
```

### First Sample Response

Retrieve a metrics configuration that filters metrics based on a specified prefix.

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 180

<?xml version="1.0" encoding="UTF-8"?>
<MetricsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>Documents</Id>
  <Filter>
    <Prefix>documents/<Prefix>
  </Filter>
</MetricsConfiguration>
```

### Second Sample Request

Retrieve a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
GET /?metrics&id=ImportantBlueDocuments HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
```

### Second Sample Response

Retrieve a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
HTTP/1.1 200 OK
```

```
x-amz-id-2: ITnGT1y4REXAMPLEPi4hkLTxouTf0hccUjo0icPEXAMPLEutBj3M7fpG1wo2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 480

<?xml version="1.0" encoding="UTF-8"?>
<MetricsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>ImportantBlueDocuments</Id>
  <Filter>
    <And>
      <Prefix>documents/</Prefix>
      <Tag>
        <Key>priority</Key>
        <Value>high</Value>
      </Tag>
      <Tag>
        <Key>class</Key>
        <Value>blue</Value>
      </Tag>
    </And>
  </Filter>
</MetricsConfiguration>
```

## Related Resources

- [PUT Bucket metrics \(p. 285\)](#)
- [DELETE Bucket metrics \(p. 90\)](#)
- [List Bucket Metrics Configurations \(p. 215\)](#)
- [Monitoring Metrics with Amazon CloudWatch in the \*Amazon Simple Storage Service Developer Guide\*.](#)

# GET Bucket notification

## Description

This implementation of the GET operation uses the notification subresource to return the notification configuration of a bucket.

If notifications are not enabled on the bucket, the operation returns an empty NotificationConfiguration element.

By default, you must be the bucket owner to read the notification configuration of a bucket. However, the bucket owner can use a bucket policy to grant permission to other users to read this configuration with the s3:GetBucketNotification permission.

For more information about setting and reading the notification configuration on a bucket, see [Setting Up Notification of Bucket Events](#). For more information about bucket policies, see [Using Bucket Policies](#).

## Requests

### Syntax

```
GET /?notification HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
CloudFunction	Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.

Name	Description
	Type: String  Ancestry: CloudFunctionConfiguration
CloudFunctionConfiguration	Container for specifying the AWS Lambda notification configuration.  Type: Container  Children: An Id, CloudFunction, and one, or more Event.  Ancestry: NotificationConfiguration
Event	Bucket event for which to send notifications.  <b>Note</b> You can add multiple instance of QueueConfiguration, TopicConfiguration, or CloudFunctionConfiguration to the notification configuration.  Type: String  Valid Values: For a list of supported event types, go to <a href="#">Configuring Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  Ancestry: TopicConfiguration and QueueConfiguration
Filter	Container for S3Key, which contains object key name filtering rules. For information about key name filtering, go to <a href="#">Configuring Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  Type: Container  Children: S3Key  Ancestor: TopicConfiguration, QueueConfiguration, or CloudFunctionConfiguration.
FilterRule	Container for key value pair that defines the criteria for the filter rule.  Container S3Key  Type: Container  Children: Name and Value  Ancestor: S3Key

Name	Description
<code>Id</code>	<p>Optional unique identifier for each of the configurations in the <code>NotificationConfiguration</code>. If you don't provide, Amazon S3 will assign an ID.</p> <p>Type: String</p> <p>Ancestry: <code>TopicConfiguration</code> and <code>QueueConfiguration</code></p>
<code>Name</code>	<p>Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to <a href="#">Configuring Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Ancestor: <code>FilterRule</code></p> <p>Valid values: <code>prefix</code> or <code>suffix</code></p>
<code>NotificationConfiguration</code>	<p>Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.</p> <p>Type: Container</p> <p>Children: one or more <code>TopicConfiguration</code>, <code>QueueConfiguration</code>, and <code>CloudFunctionConfiguration</code> elements.</p> <p>Ancestry: None</p>
<code>Queue</code>	<p>Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type.</p> <p>Type: String</p> <p>Ancestry: <code>TopicConfiguration</code></p>
<code>QueueConfiguration</code>	<p>Container for specifying a configuration when you want Amazon S3 to publish events to an Amazon Simple Queue Service (Amazon SQS) queue.</p> <p>Type: Container</p> <p>Children: An <code>Id</code>, <code>Topic</code>, and one, or more <code>Event</code>.</p> <p>Ancestry: <code>NotificationConfiguration</code></p>

Name	Description
S3Key	<p>Container for object key name prefix and suffix filtering rules.</p> <p>Type: Container</p> <p>Children: One or more <code>FilterRule</code></p> <p>Ancestor: <code>Filter</code></p>
Topic	<p>Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.</p> <p>Type: String</p> <p>Ancestry: <code>TopicConfiguration</code></p>
TopicConfiguration	<p>Container for specifying the configuration when you want Amazon S3 to publish events to an Amazon Simple Notification Service (Amazon SNS) topic.</p> <p>Type: Container</p> <p>Children: An <code>Id</code>, <code>Topic</code>, and one, or more <code>Event</code>.</p> <p>Ancestry: <code>NotificationConfiguration</code></p>
Value	<p>Specifies the object key name prefix or suffix to filter on.</p> <p>Type: String</p> <p>Ancestor: <code>FilterRule</code></p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request returns the notification configuration on the bucket quotes.s3.amazonaws.com.

```
GET ?notification HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 15 Oct 2014 16:59:03 GMT
Authorization: authorization string
```

### Sample Response

This response returns that the notification configuration for the specified bucket.

```
HTTP/1.1 200 OK
```

```
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Wed, 15 Oct 2014 16:59:04 GMT
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TopicConfiguration>
    <Id>YjVkJM2Y0YmUtNGI3NC00ZjQyLWEwNGItNDIyYWUxY2I0N2M4</Id>
    <Topic>arn:aws:sns:us-east-1:account-id:s3notificationtopic2</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
    <Event>s3:ObjectCreated:*</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

## Related Resources

- [PUT Bucket notification \(p. 290\)](#)

# GET Bucket object lock configuration

Service: Amazon Simple Storage Service

Gets the Object Lock configuration for a bucket. The rule specified in the Object Lock configuration will be applied by default to every new object placed in the specified bucket.

## Request Syntax

```
GET /?object-lock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests \(AWS Signature Version 4\))
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request does not have a request body.

## Response Syntax

```
<ObjectLockConfiguration>
  <ObjectLockEnabled>string</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode>string</Mode>
      <Years>integer</Years>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

### [ObjectLockConfiguration \(p. 169\)](#)

Root level tag for the ObjectLockConfiguration parameters.

### [ObjectLockEnabled \(p. 169\)](#)

Indicates whether this bucket has an Object Lock configuration enabled.

Type: String

Valid Values: Enabled

### [Rule \(p. 169\)](#)

The Object Lock rule that will be applied to objects placed in this bucket.

Type: [ObjectLockRule \(p. 332\)](#) object

## Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

# GET BucketPolicyStatus

## Description

This operation retrieves the policy status for an Amazon S3 bucket, indicating whether the bucket is public. In order to use this operation, you must have the `s3:GetBucketPolicyStatus` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information about when Amazon S3 considers a bucket public, see [The Meaning of "Public"](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /<bucket-name>?policyStatus HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

Name	Description
PolicyStatus	<p>Container element for bucket policy status.</p> <p>Type: Container</p> <p>Children: IsPublic</p>
IsPublic	<p>Indicates whether this bucket currently has a public access policy.</p> <p>Type: Boolean</p> <p>Ancestor: PolicyStatus</p> <p>Valid values: TRUE   FALSE</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request gets a bucket policy status.

```
GET /<bucket-name>?policyStatus HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0

<PolicyStatus>
  <IsPublic>TRUE</IsPublic>
</PolicyStatus>
```

## Related Resources

- [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service Developer Guide*.
- [GET PublicAccessBlock \(p. 153\)](#)
- [PUT PublicAccessBlock \(p. 277\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)

- [GET PublicAccessBlock \(p. 69\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)
- [DELETE PublicAccessBlock \(p. 68\)](#)

# GET Bucket Object versions

## Description

You can use the `versions` subresource to list metadata about all of the versions of objects in a bucket. You can also use request parameters as selection criteria to return metadata about a subset of all the object versions. For more information, see [Request Parameters \(p. 173\)](#).

### Note

A 200 `OK` response can contain valid or invalid XML. Make sure to design your application to parse the contents of the response and handle it appropriately.

To use this operation, you must have `READ` access to the bucket.

## Requests

### Syntax

```
GET /?versions HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

## Request Parameters

This implementation of `GET` uses the parameters in the following table to return a subset of the objects in a bucket.

Parameter	Description	Required
<code>delimiter</code>	<p>A delimiter is a character that you specify to group keys. All keys that contain the same string between the <code>prefix</code> and the first occurrence of the delimiter are grouped under a single result element in <code>CommonPrefixes</code>. These groups are counted as one result against the <code>max-keys</code> limitation. These keys are not returned elsewhere in the response. Also, see <code>prefix</code>.</p> <p>Type: String</p> <p>Default: None</p>	No
<code>encoding-type</code>	<p>Requests Amazon S3 to encode the response and specifies the encoding method to use.</p> <p>An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</p> <p>Type: String</p> <p>Default: None</p>	No

Parameter	Description	Required
	Valid value: url	
key-marker	<p>Specifies the key in the bucket that you want to start listing from. Also, see <code>version-id-marker</code>.</p> <p>Type: String</p> <p>Default: None</p>	No
max-keys	<p>Sets the maximum number of keys returned in the response body. The response might contain fewer keys, but will never contain more. If additional keys satisfy the search criteria, but were not returned because <code>max-keys</code> was exceeded, the response contains <code>&lt;isTruncated&gt;true&lt;/isTruncated&gt;</code>. To return the additional keys, see <code>key-marker</code> and <code>version-id-marker</code>.</p> <p>Type: String</p> <p>Default: 1000</p>	No
prefix	<p>Use this parameter to select only those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using <code>prefix</code> to make groups in the same way you'd use a folder in a file system.) You can use <code>prefix</code> with <code>delimiter</code> to roll up numerous objects into a single result under <code>CommonPrefixes</code>. Also, see <code>delimiter</code>.</p> <p>Type: String</p> <p>Default: None</p>	No
version-id-marker	<p>Specifies the object version you want to start listing from. Also, see <code>key-marker</code>.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Values: Valid version ID   Default</p> <p>Constraint: May not be an empty string</p>	No

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

Name	Description
DeleteMarker	<p>Container for an object that is a delete marker.</p> <p>Type: Container</p> <p>Children: Key, VersionId, IsLatest, LastModified, Owner</p> <p>Ancestor: ListVersionsResult</p>
DisplayName	<p>Object owner's name.</p> <p><b>Important</b> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p> <p>Type: String</p> <p>Ancestor: ListVersionsResult.Version.Owner   ListVersionsResult.DeleteMarker.Owner</p>
Encoding-Type	<p>Encoding type used by Amazon S3 to encode object key names in the XML response.</p> <p>If you specify encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements:</p> <p><code>KeyMarker</code>, <code>NextKeyMarker</code>, <code>Prefix</code>, <code>Key</code>, and <code>Delimiter</code>.</p> <p>Type: String</p> <p>Ancestor: ListBucketResult</p>
ETag	<p>The entity tag is an MD5 hash of the object. The ETag only reflects changes to the contents of an object, not its metadata.</p> <p>Type: String</p> <p>Ancestor: ListVersionsResult.Version</p>
ID	<p>Object owner's ID.</p> <p>Type: String</p> <p>Ancestor: ListVersionsResult.Version.Owner   ListVersionsResult.DeleteMarker.Owner</p>
IsLatest	<p>Specifies whether the object is (<code>true</code>) or is not (<code>false</code>) the current version of an object.</p> <p>Type: Boolean</p>

Name	Description
	<p>Valid Values: <code>true</code>   <code>false</code></p> <p>Ancestor: <code>ListVersionsResult.Version</code>   <code>ListVersionsResult.DeleteMarker</code></p>
<code>IsTruncated</code>	<p>A flag that indicates whether (<code>true</code>) or not (<code>false</code>) Amazon S3 returned all of the results that satisfied the search criteria. If your results were truncated, you can make a follow-up paginated request using the <code>NextKeyMarker</code> and <code>NextVersionIdMarker</code> response parameters as a starting place in another request to return the rest of the results.</p> <p>Type: Boolean</p> <p>Valid Values: <code>true</code>   <code>false</code></p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>Key</code>	<p>The object's key.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult.Version</code>   <code>ListVersionsResult.DeleteMarker</code></p>
<code>KeyMarker</code>	<p>Marks the last <code>Key</code> returned in a truncated response.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>LastModified</code>	<p>Date and time the object was last modified.</p> <p>Type: Date</p> <p>Ancestor: <code>ListVersionsResult.Version</code>   <code>ListVersionsResult.DeleteMarker</code></p>
<code>ListVersionsResult</code>	<p>Container for the result.</p> <p>Type: Container</p> <p>Children: All elements in the response</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>MaxKeys</code>	<p>Specifies the maximum number of objects to return.</p> <p>Type: String</p> <p>Default: 1000</p> <p>Valid Values: Integers from 1 to 1000, inclusive</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>Name</code>	<p>Bucket owner's name.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult</code></p>

Name	Description
<code>NextKeyMarker</code>	<p>When the number of responses exceeds the value of <code>MaxKeys</code>, <code>NextKeyMarker</code> specifies the first key not returned that satisfies the search criteria. Use this value for the <code>key-marker</code> request parameter in a subsequent request.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>NextVersionIdMarker</code>	<p>When the number of responses exceeds the value of <code>MaxKeys</code>, <code>NextVersionIdMarker</code> specifies the first object version not returned that satisfies the search criteria. Use this value for the <code>version-id-marker</code> request parameter in a subsequent request.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>Owner</code>	<p>Bucket owner.</p> <p>Type: String</p> <p>Children: <code>DisplayName, ID</code></p> <p>Ancestor: <code>ListVersionsResult.Version   ListVersionsResult.DeleteMarker</code></p>
<code>Prefix</code>	<p>Selects objects that start with the value supplied by this parameter.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>Size</code>	<p>Size in bytes of the object.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult.Version</code></p>
<code>StorageClass</code>	<p>Always STANDARD.</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult.Version</code></p>
<code>Version</code>	<p>Container for version information.</p> <p>Type: Container</p> <p>Ancestor: <code>ListVersionsResult</code></p>
<code>VersionId</code>	<p>Version ID of an object</p> <p>Type: String</p> <p>Ancestor: <code>ListVersionsResult.Version   ListVersionsResult.DeleteMarker</code></p>

Name	Description
VersionIdMarker	Marks the last version of the key returned in a truncated response.  Type: String  Ancestor: ListVersionsResult

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns all of the versions of all of the objects in the specified bucket.

```
GET /?versions HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Sample Response to GET Versions

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

<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <Name>bucket</Name>
  <Prefix>my</Prefix>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>5</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>my-image.jpg</Key>
    <VersionId>3/L4kqtJl40Nr8X8gdRQBpUMLUo</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Version>
  <DeleteMarker>
    <Key>my-second-image.jpg</Key>
    <VersionId>03jpff543dhffds434rfdsFDN943fdsFkdmqn892</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-11-12T17:50:30.000Z</LastModified>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </DeleteMarker>
</ListVersionsResult>
```

```

        </Owner>
    </DeleteMarker>
    <Version>
        <Key>my-second-image.jpg</Key>
        <VersionId>QUpfdndhfd8438MNFDN93jdnJFkdmqnh893</VersionId>
        <IsLatest>false</IsLatest>
        <LastModified>2009-10-10T17:50:30.000Z</LastModified>
        <ETag>&quot;9b2cf535f27731c974343645a3985328&quot;</ETag>
        <Size>166434</Size>
        <StorageClass>STANDARD</StorageClass>
        <Owner>
            <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeefbf76c078efc7c6caea54ba06a</ID>
            <DisplayName>mtd@amazon.com</DisplayName>
        </Owner>
    </Version>
    <DeleteMarker>
        <Key>my-third-image.jpg</Key>
        <VersionId>03jpff543dhffds434rfdsFDN943fdsFkdmqnh892</VersionId>
        <IsLatest>true</IsLatest>
        <LastModified>2009-10-15T17:50:30.000Z</LastModified>
        <Owner>
            <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeefbf76c078efc7c6caea54ba06a</ID>
            <DisplayName>mtd@amazon.com</DisplayName>
        </Owner>
    </DeleteMarker>
    <Version>
        <Key>my-third-image.jpg</Key>
        <VersionId>UIORUnfndhnw89493jJFJ</VersionId>
        <IsLatest>false</IsLatest>
        <LastModified>2009-10-11T12:50:30.000Z</LastModified>
        <ETag>&quot;772cf535f27731c974343645a3985328&quot;</ETag>
        <Size>64</Size>
        <StorageClass>STANDARD</StorageClass>
        <Owner>
            <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeefbf76c078efc7c6caea54ba06a</ID>
            <DisplayName>mtd@amazon.com</DisplayName>
        </Owner>
    </Version>
</ListVersionsResult>

```

## Sample Request

The following request returns objects in the order they were stored, returning the most recently stored object first starting with the value for key-marker.

```

GET /?versions&key-marker=key2 HTTP/1.1
Host: s3.amazonaws.com
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, /*
Date: Thu, 10 Dec 2009 22:46:32 +0000
Authorization: signatureValue

```

## Sample Response

```

<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Name>mtp-versioning-fresh</Name>
    <Prefix/>
    <KeyMarker>key2</KeyMarker>
    <VersionIdMarker/>
    <MaxKeys>1000</MaxKeys>

```

```

<IsTruncated>false</IsTruncated>
<Version>
  <Key>key3</Key>
  <VersionId>I5VhmK6CDDdQ5PwfelgcHZWmHDpcv7gfmfc29UBxsKU.</VersionId>
  <IsLatest>true</IsLatest>
  <LastModified>2009-12-09T00:19:04.000Z</LastModified>
  <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
  <Size>217</Size>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
</Version>
<DeleteMarker>
  <Key>sourcekey</Key>
  <VersionId>qDhprLU80sAlCFLu2DWgXAEDgKzWarn-HS_JU0TvYqs.</VersionId>
  <IsLatest>true</IsLatest>
  <LastModified>2009-12-10T16:38:11.000Z</LastModified>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
  </Owner>
</DeleteMarker>
<Version>
  <Key>sourcekey</Key>
  <VersionId>wxxQ7ezLaL5JN2Sis1q66Syxxo0k7uHTUpb9qiimxNg.</VersionId>
  <IsLatest>false</IsLatest>
  <LastModified>2009-12-10T16:37:44.000Z</LastModified>
  <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
  <Size>217</Size>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
</Version>
</ListVersionsResult>

```

## Sample Request Using prefix

This example returns objects whose keys begin with source.

```

GET /?versions&prefix=source HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string

```

## Sample Response

```

<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>mtp-versioning-fresh</Name>
  <Prefix>source</Prefix>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <DeleteMarker>
    <Key>sourcekey</Key>
    <VersionId>qDhprLU80sAlCFLu2DWgXAEDgKzWarn-HS_JU0TvYqs.</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-10T16:38:11.000Z</LastModified>
    <Owner>

```

```
<ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
</Owner>
</DeleteMarker>
<Version>
<Key>sourcekey</Key>
<VersionId>wxxQ7ezLaL5JN2Sis1q66Syxxo0k7uHTUpb9qiimxNg.</VersionId>
<IsLatest>false</IsLatest>
<LastModified>2009-12-10T16:37:44.000Z</LastModified>
<ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
<Size>217</Size>
<Owner>
<ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
</Owner>
<StorageClass>STANDARD</StorageClass>
</Version>
</ListVersionsResult>
```

## Sample Request Using key-marker and version-id-marker Parameters

The following example returns objects starting at the specified key (`key-marker`) and version ID (`version-id-marker`).

```
GET /?versions&key-marker=key3&version-id-marker=t46ZenlyTZBnj HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: signatureValue
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>mtp-versioning-fresh</Name>
<Prefix/>
<KeyMarker>key3</KeyMarker>
<VersionIdMarker>t46ZenlyTZBnj</VersionIdMarker>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<DeleteMarker>
<Key>sourcekey</Key>
<VersionId>qDhprLU80sAlCFLu2DWgXAEDgKzWarn-HS_JU0TvYqs.</VersionId>
<IsLatest>true</IsLatest>
<LastModified>2009-12-10T16:38:11.000Z</LastModified>
<Owner>
<ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
</Owner>
</DeleteMarker>
<Version>
<Key>sourcekey</Key>
<VersionId>wxxQ7ezLaL5JN2Sis1q66Syxxo0k7uHTUpb9qiimxNg.</VersionId>
<IsLatest>false</IsLatest>
<LastModified>2009-12-10T16:37:44.000Z</LastModified>
<ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
<Size>217</Size>
<Owner>
<ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
</Owner>
<StorageClass>STANDARD</StorageClass>
</Version>
</ListVersionsResult>
```

## Sample Request Using key-marker, version-id-marker and max-keys

The following request returns up to three (the value of `max-keys`) objects starting with the key specified by `key-marker` and the version ID specified by `version-id-marker`.

```
GFT /?versions&key-marker=key3&version-id-marker=t46Z0menLYTZBnj HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: authorization string
```

## Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>null</VersionIdMarker>
  <NextKeyMarker>key3</NextKeyMarker>
  <NextVersionIdMarker>d-d309mfjFrUmoQ0DBsVqmcMV15OI.</NextVersionIdMarker>
  <MaxKeys>2</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>8XECiENpj8pydEDJdd-_VRrvaGKAHOaGMNW7tg6UVii.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-09T00:18:23.000Z</LastModified>
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faebf76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
  <Version>
    <Key>key3</Key>
    <VersionId>d-d309mfjFri40QYukDozqBt3UmoQ0DBsVqmcMV15OI.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-09T00:18:08.000Z</LastModified>
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
    <Size>217</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faebf76c078efc7c6caea54ba06a</ID>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>
```

## Sample Request Using the Delimiter and the Prefix Parameters

Assume you have the following keys in your bucket, `example-bucket`.

`photos/2006/January/sample.jpg`

`photos/2006/February/sample.jpg`

`photos/2006/March/sample.jpg`

videos/2006/March/sample.wmv

sample.jpg

The following GET versions request specifies the delimiter parameter with value "/".

```
GET /?versions&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 02 Feb 2011 20:34:56 GMT
Authorization: authorization string
```

The list of keys from the specified bucket are shown in the following response.

The response returns the sample.jpg key in a <Version> element. However, because all the other keys contain the specified delimiter, a distinct substring, from the beginning of the key to the first occurrence of the delimiter, from each of these keys is returned in a <CommonPrefixes> element. The key substrings, photos/ and videos/, in the <CommonPrefixes> element indicate that there are one or more keys with these key prefixes.

This is a useful scenario if you use key prefixes for your objects to create a logical folder like structure. In this case you can interpret the result as the folders photos/ and videos/ have one or more objects.

```
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Name>mvbucketwithversionon1</Name>
<Prefix></Prefix>
<KeyMarker></KeyMarker>
<VersionIdMarker></VersionIdMarker>
<MaxKeys>1000</MaxKeys>
<Delimiter>/</Delimiter>
<IsTruncated>false</IsTruncated>

<Version>
<Key>Sample.jpg</Key>
<VersionId>t0xMzQ1BsGyGCz1YuMWMP90cdXLzqOCH</VersionId>
<IsLatest>true</IsLatest>
<LastModified>2011-02-02T18:46:20.000Z</LastModified>
<ETag>"3305f2cfc46c0f04559748bb039d69ae"</ETag>
<Size>3191</Size>
<Owner>
<ID>852b113e7a2f25102679df27bb0ae12b3f85be6f290b936c4393484be31bebcc</ID>
<DisplayName>display-name</DisplayName>
</Owner>
<StorageClass>STANDARD</StorageClass>
</Version>

<CommonPrefixes>
<Prefix>photos/</Prefix>
</CommonPrefixes>
<CommonPrefixes>
<Prefix>videos/</Prefix>
</CommonPrefixes>
</ListVersionsResult>
```

In addition to the delimiter parameter you can filter results by adding a prefix parameter as shown in the following request.

```
GET /?versions&prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 02 Feb 2011 19:34:02 GMT
Authorization: authorization string
```

In this case the response will include only objects keys that start with the specified prefix. The value returned in the <CommonPrefixes> element is a substring from the beginning of the key to the first occurrence of the specified delimiter after the prefix.

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>example-bucket</Name>
  <Prefix>photos/2006/</Prefix>
  <KeyMarker></KeyMarker>
  <VersionIdMarker></VersionIdMarker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>photos/2006/</Key>
    <VersionId>3U275dAA4gz8Z0qOPHtJCUUi60krpCdy</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2011-02-02T18:47:27.000Z</LastModified>
    <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
    <Size>0</Size>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>display-name</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/March/</Prefix>
  </CommonPrefixes>
</ListVersionsResult>
```

## Related Resources

- [GET Bucket Objects \(p. 111\)](#)
- [GET Object \(p. 349\)](#)
- [PUT Object \(p. 412\)](#)
- [DELETE Object \(p. 343\)](#)

# GET Bucket policy

## Description

This implementation of the `GET` operation uses the `policy` subresource to return the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the `GetBucketPolicy` permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have `GetBucketPolicy` permissions, Amazon S3 returns a `403 Access Denied` error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a `405 Method Not Allowed` error.

### Important

As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see [Using Bucket Policies and User Policies](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

The response contains the (JSON) policy of the specified bucket.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the policy of the specified bucket.

```
GET ?policy HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByru9pO4SAMPLEAtRPfTaOFG==
x-amz-request-id: 656c76696e67SAMPLE57374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3

{
    "Version": "2008-10-17",
    "Id": "aaaa-bbbb-cccc-dddd",
    "Statement": [
        {
            "Effect": "Deny",
            "Sid": "1",
            "Principal": {
                "AWS": ["111122223333", "444455556666"]
            },
            "Action": ["s3:*"],
            "Resource": "arn:aws:s3:::bucket/*"
        }
    ]
}
```

## Related Resources

- [GET Bucket Objects \(p. 111\)](#)

# GET Bucket replication

## Description

Returns a bucket's replication configuration.

### Note

It can take a while for `PUT Bucket replication` and `DELETE Bucket replication` requests to fully propagate. If you submit a `GET Bucket replication` request soon after submitting either of those requests, might not return the latest replication configuration.

For information about replication configuration, see [Cross-Region Replication \(CRR\)](#) in the *Amazon Simple Storage Service Developer Guide*.

This operation requires permissions for the `s3:GetReplicationConfiguration` action. For more information about permissions, see [Using Bucket Policies and User Policies](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?replication HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string
```

For more information about authorization, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of GET returns the following response elements.

Element	Description
<code>ReplicationConfiguration</code>	<p>The container for replication rules.</p> <p>Type: Container</p> <p>Children: <code>Rule</code></p> <p>Ancestor: None</p>
<code>Rule</code>	<p>The container for information about a particular replication rule.</p> <p>Type: Container</p> <p>Ancestor: <code>ReplicationConfiguration</code></p>
<code>Role</code>	<p>The Amazon Resource Name (ARN) of an AWS Identity and Access Management (IAM) role that Amazon S3 assumes when replicating objects.</p> <p>Type: String</p> <p>Ancestor: <code>Rule</code></p>
<code>ID</code>	<p>The unique identifier for the rule.</p> <p>Type: String</p> <p>Ancestor: <code>Rule</code></p>
<code>Status</code>	<p>Whether a rule is enabled. If <code>Status</code> is not set to <code>Enabled</code>, Amazon S3 ignores the rule</p> <p>Type: String</p> <p>Ancestor: <code>Rule</code></p> <p>Valid values: <code>Enabled</code>, <code>Disabled</code>.</p>
<code>Prefix</code>	<p>The object key name prefix that identifies the objects that the rule applies to.</p> <p><b>Note</b> If the replication configuration uses the <code>Filter</code> element instead of <code>Prefix</code>, Amazon S3 returns the <code>Filter</code> element. For more information about the <code>Filter</code> element, see the next table.</p> <p>Type: String</p> <p>Ancestor: <code>Rule</code></p>
<code>Destination</code>	<p>A container for information about the destination.</p> <p>Type: Container</p> <p>Ancestor: <code>Rule</code></p>
<code>Account</code>	The account ID of the owner of the destination bucket. In a cross-account scenario, if you tell Amazon S3 to change

Element	Description
	<p>replica ownership to the AWS account that owns the destination bucket, this is the account ID of the owner of the destination bucket. For more information, see <a href="#">Cross-Region Replication Additional Configuration: Change Replica Owner</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>If the owner override option is not set in a replication configuration, the response does include this element.</p> <p>Type: String</p> <p>Ancestor: Destination</p>
Bucket	<p>The name of the bucket where Amazon S3 stores replicas of objects identified by the rule.</p> <p>Type: String</p> <p>Ancestor: Destination</p>
StorageClass	<p>The storage class for replicated objects. This field is returned only if you set the storage class when you configured cross-region replication (with <a href="#">PUT Bucket replication (p. 302)</a>).</p> <p>Type: String</p> <p>Ancestor: Destination</p>
AccessControlTranslation	<p>If you set the owner override option in the replication configuration, Amazon S3 returns this element. It identifies the owner of the replicas.</p> <p>If this element isn't present, replicas are owned by the same AWS account that owns the source object.</p> <p>Type: String</p> <p>Ancestor: Destination</p>
Owner	<p>Identifies the owner of the replicas. Amazon S3 returns this element only if you configured owner override option, in a cross-account scenario.</p> <p>Type: String</p> <p>Ancestor: AccessControlTranslation</p>

## Rule Filter Response Elements

A replication configuration rule can specify a filter to identify a subset of source objects to apply the rule to. The response can return the following additional elements, which are related to filtering.

Element	Description
<code>Filter</code>	The container that describes the filters used to identify the source objects that you want to replicate.  Ancestor: <code>Rule</code>
<code>And</code>	The container for the <code>Prefix</code> and one or more <code>Tag</code> elements. If the <code>And</code> element is present, it includes at least one child element.  Ancestor: <code>Filter</code>
<code>Prefix</code>	The object key prefix that identifies one or more objects that the rule applies to.  <b>Note</b> The earlier version of replication configuration (V1) supported only the key prefix as a rule filter. In V1, the response returns the <code>Prefix</code> element as a child of the <code>Rule</code> element. Amazon S3 supports this behavior for backward compatibility. For more information, see <a href="#">Backward Compatibility</a> in the <i>Amazon S3 Developer Guide</i> .  Type: String  Ancestor: <code>Filter</code> , or <code>And</code> (if present), or <code>Rule</code> (if you are using the earlier version of replication configuration).
<code>Tag</code>	A container that provides a tag key and value.  Ancestor: <code>Filter</code> or <code>And</code> (if present)
<code>Key</code>	A tag key.  Type: String  Ancestor: <code>Tag</code>
<code>Value</code>	A tag value.  Type: String  Ancestor: <code>Tag</code>

If you include the `Filter` element in a replication configuration, you must also include the `DeleteMarkerReplication` and `Priority` elements. The response also returns those elements.

Element	Description
<code>DeleteMarkerReplication</code>	A container that describes whether Amazon S3 replicates the delete markers.  Ancestor: <code>Rule</code>
<code>Status</code>	Indicates whether to replicate delete markers.

Element	Description
	Type: String  Ancestor: <code>DeleteMarkerReplication</code>
<code>Priority</code>	If you specify multiple rules with overlapping filters, identifies the rule priority. For example, if two rules apply to the same object based on the <code>Filter</code> specified, then the rule with higher priority supersedes. The higher the numerical value of this element, the higher the rule priority. For more information, see <a href="#">Backward Compatibility</a> in the <i>Amazon S3 Developer Guide</i> .  Type: Integer  Ancestor: <code>Rule</code>

## Encryption Response Elements

If a replication configuration specifies replicating objects created with server-side encryption using an AWS KMS-managed key, the response returns the following additional elements. For more information, see [CRR: Replicating Objects Created with SSE Using AWS KMS-Managed Encryption Keys](#) in the *Amazon Simple Storage Service Developer Guide*.

Element	Description
<code>SourceSelectionCriteria</code>	A container that describes additional filters that identify the source objects that you want to replicate.  Type: String  Ancestor: <code>Rule</code>
<code>SseKmsEncryptedObjects</code>	A container for the <code>Status</code> element.  Type: String  Ancestor: <code>SourceSelectionCriteria</code>
<code>Status</code>	A flag that tells Amazon S3 whether to replicate objects created with server-side encryption using an AWS KMS-managed key.  Type: String  Ancestor: <code>SseKmsEncryptedObjects</code>
<code>EncryptionConfiguration</code>	A container that provides information about encryption.  Type: String  Ancestor: <code>Destination</code>
<code>ReplicaKmsKeyID</code>	The AWS KMS Key ID—the Key Amazon Resource Name (ARN) or Alias ARN—of the destination bucket. Amazon S3 uses this key to encrypt replicas.  Type: String

Element	Description
	Ancestor: <code>EncryptionConfiguration</code>

## Special Errors

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
<code>NoSuchReplicationConfiguration</code>	There is no replication configuration with that name.	404 Not Found	Client

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Retrieve Replication Configuration Information

The following GET request retrieves information about the replication configuration set for the `examplebucket` bucket:

```
GET /?replication HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 10 Feb 2015 00:17:21 GMT
Authorization: authorization string
```

The following response shows that replication is enabled on the bucket. The empty prefix indicates that Amazon S3 will replicate all objects that are created in the `examplebucket` bucket. The `Destination` element identifies the target bucket where Amazon S3 creates the object replicas, and the storage class (`STANDARD_IA`) that Amazon S3 uses when creating replicas.

Amazon S3 assumes the specified IAM role to replicate objects on behalf of the bucket owner, which is the AWS account that created the bucket.

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hk1TXouTf0hccUjo0iCPjz6FnfIutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991C342example
Date: Tue, 10 Feb 2015 00:17:23 GMT
Server: AmazonS3
Content-Length: contentlength

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Role>arn:aws:iam::35667example:role/CrossRegionReplicationRoleForS3</Role>
  <Rule>
    <ID>rule1</ID>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <And>
        <Prefix>TaxDocs</Prefix>
        <Tag>
```

```
<Key>key1</Key>
<Value>value1</Value>
</Tag>
<Tag>
<Key>key1</Key>
<Value>value1</Value>
</Tag>
</And>
</Filter>
<Destination>
<Bucket>arn:aws:s3:::exampletargetbucket</Bucket>
</Destination>
</Rule>
</ReplicationConfiguration>
```

## Related Resources

- [PUT Bucket replication \(p. 302\)](#)
- [DELETE Bucket replication \(p. 95\)](#)

# GET Bucket requestPayment

## Description

This implementation of the GET operation uses the `requestPayment` subresource to return the request payment configuration of a bucket. To use this version of the operation, you must be the bucket owner. For more information, see [Requester Pays Buckets](#).

## Requests

### Syntax

```
GET ?requestPayment HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
Payer	Specifies who pays for the download and request fees.  Type: Enum  Valid Values: Requester   BucketOwner  Ancestor: RequestPaymentConfiguration
RequestPaymentConfiguration	Container for Payer.  Type: Container

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the payer for the bucket, `colorpictures`.

```
GET ?requestPayment HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: 0
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

This response shows that the bucket is a Requester Pays bucket, meaning the person requesting a download from this bucket pays the transfer fees.

### Related Resources

- [GET Bucket \(List Objects\) Version 1 \(p. 111\)](#)

# GET Bucket tagging

## Description

This implementation of the GET operation uses the tagging subresource to return the tag set associated with the bucket.

To use this operation, you must have permission to perform the `s3:GetBucketTagging` action. By default, the bucket owner has this permission and can grant this permission to others.

## Requests

### Syntax

```
GET /?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
Tagging	Contains the TagSet and Tag elements.  Type: Container  Ancestry: None

Name	Description
TagSet	Contains the tag set.  Type: Container  Ancestry: Tagging
Tag	Contains the tag information.  Type: Container  Ancestry: TagSet
Key	Name of the tag  Type: String  Ancestry: Tag
Value	Value of the tag  Type: String  Ancestry: Tag

## Special Errors

- **NoSuchTagsetError** - There is no tag set associated with the bucket.

## Examples

### Sample Request

The following request returns the tag set of the specified bucket.

```
GET ?tagging HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2009 12:00:00 GMT
Connection: close
Server: AmazonS3

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
```

```
<Value>jsmith</Value>
</Tag>
</TagSet>
</Tagging>
```

## Related Resources

- [PUT Bucket tagging \(p. 314\)](#)
- [DELETE Bucket tagging \(p. 97\)](#)

# GET Bucket versioning

## Description

This implementation of the `GET` operation uses the `versioning` subresource to return the versioning state of a bucket. To retrieve the versioning state of a bucket, you must be the bucket owner.

This implementation also returns the MFA Delete status of the versioning state, i.e., if the MFA Delete status is enabled, the bucket owner must use an authentication device to change the versioning state of the bucket.

There are three versioning states:

- If you enabled versioning on a bucket, the response is:

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

- If you suspended versioning on a bucket, the response is:

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Suspended</Status>
</VersioningConfiguration>
```

- If you never enabled (or suspended) versioning on a bucket, the response is:

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/" />
```

## Requests

### Syntax

```
GET /?versioning HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of `GET` returns the following response elements.

Name	Description
<code>MfaDelete</code>	<p>Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with <code>MfaDelete</code>. If the bucket has never been so configured, this element is not returned.</p> <p>Type: Enum</p> <p>Valid Values: Disabled   Enabled</p> <p>Ancestor: <code>VersioningConfiguration</code></p>
<code>Status</code>	<p>The versioning state of the bucket.</p> <p>Type: Enum</p> <p>Valid Values: Suspended   Enabled</p> <p>Ancestor: <code>VersioningConfiguration</code></p>
<code>VersioningConfiguration</code>	<p>Container for the <code>Status</code> response element.</p> <p>Type: Container</p> <p>Ancestor: None</p>

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This example returns the versioning state of `myBucket`.

```
GET /?versioning HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

## Sample Response

The following is a sample of the response body (only) that shows bucket versioning is enabled.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

## Related Resources

- [GET Object \(p. 349\)](#)
- [PUT Object \(p. 412\)](#)
- [DELETE Object \(p. 343\)](#)

# GET Bucket website

## Description

This implementation of the `GET` operation returns the website configuration associated with a bucket. To host website on Amazon S3, you can configure a bucket as website by adding a website configuration. For more information about hosting websites, go to [Hosting Websites on Amazon S3](#) in the *Amazon Simple Storage Service Developer Guide*.

This `GET` operation requires the `S3:GetBucketWebsite` permission. By default, only the bucket owner can read the bucket website configuration. However, bucket owners can allow other users to read the website configuration by writing a bucket policy granting them the `S3:GetBucketWebsite` permission.

## Requests

### Syntax

```
GET /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

The response XML includes same elements that were uploaded when you configured the bucket as website. For more information, see [PUT Bucket website \(p. 321\)](#).

## Examples

### Sample Request

This request retrieves website configuration on the specified bucket.

```
GET ?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Thu, 27 Jan 2011 00:49:20 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:n0Nhek72Ufg/u7Sm5C1dqRLs8XX=
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 3848CD259D811111
Date: Thu, 27 Jan 2011 00:49:26 GMT
Content-Length: 240
Content-Type: application/xml
Transfer-Encoding: chunked
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>404.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

## Related Resources

- [DELETE Bucket website \(p. 99\)](#)
- [PUT Bucket website \(p. 321\)](#)

# HEAD Bucket

## Description

This operation is useful to determine if a bucket exists and you have permission to access it. The operation returns a 200 OK if the bucket exists and you have permission to access it. Otherwise, the operation might return responses such as 404 Not Found and 403 Forbidden.

To use this operation, you must have permissions to perform the s3:ListBucket action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
HEAD / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Elements

This implementation of the operation does not use request elements.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

```
HEAD / HTTP/1.1
Date: Fri, 10 Feb 2012 21:34:55 GMT
Authorization: authorization string
Host: myawsbucket.s3.amazonaws.com
Connection: Keep-Alive
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagkKdBX2z7I1/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Fri, 10 2012 21:34:56 GMT
Server: AmazonS3
```

# List Bucket Analytics Configurations

## Description

This implementation of the `GET` operation returns a list of analytics configurations for the bucket. You can have up to 1,000 analytics configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. You should always check the `IsTruncated` element in the response. If there are no more configurations to list, `IsTruncated` is set to false. If there are more configurations to list, `IsTruncated` is set to true, and there will be a value in `NextContinuationToken`. You use the `NextContinuationToken` value to continue the pagination of the list by passing the value in `continuation-token` in the request to `GET` the next page.

To use this operation, you must have permissions to perform the `s3:GetAnalyticsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about Amazon S3 analytics feature, see [Amazon S3 Analytics – Storage Class Analysis](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?analytics HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of `GET` uses the parameters in the following table.

Parameter	Description	Required
<code>continuation-token</code>	When the Amazon S3 response to this API call is truncated (that is, when the <code>IsTruncated</code> response element value is true), the response also includes the <code>NextContinuationToken</code> element, the value of which you can use in the next request as the <code>continuation-token</code> to list the next page. The continuation token is an opaque value that Amazon S3 understands. Type: String  Default: None	No

### Request Elements

This implementation of the operation does not use request elements.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
ContinuationToken	The marker that is used as a starting point for this analytics configuration list response. This value is present if it was sent in the request.  Type: String  Ancestor: <code>ListBucketAnalyticsConfigurationsResult</code>
IsTruncated	Indicates whether the returned list of analytics configurations is complete. A value of true indicates that the list is not complete and the <code>NextContinuationToken</code> is provided for a subsequent request.  Type: Boolean  Ancestor: <code>ListAnalyticsConfigurationsResult</code>
AnalyticsConfiguration	Contains the analytics configuration. For the XML structure, see <a href="#">GET Bucket analytics (p. 126)</a> .  Type: Container  Ancestor: <code>ListAnalyticsConfigurationsResult</code>
ListAnalyticsConfigurationsResult	The list of analytics configurations for a bucket.  Type: Container
NextContinuationToken	The marker used to continue an analytics configuration listing that has been truncated. Use the <code>NextContinuationToken</code> from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.  Type: String  Ancestor: <code>ListBucketAnalyticsConfigurationsResult</code>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Listing Analytics Configurations

The following request returns the analytics configurations in example-bucket.

#### Sample Request

```
GET /?analytics HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXR3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Length: length
Server: AmazonS3

<ListBucketAnalyticsConfigurationResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <AnalyticsConfiguration>
    <Id>list1</Id>
    <Filter>
      <And>
        <Prefix>images/</Prefix>
        <Tag>
          <Key>dog</Key>
          <Value>corgi</Value>
        </Tag>
      </And>
    </Filter>
    <StorageClassAnalysis>
      <DataExport>
        <OutputSchemaVersion>V_1</OutputSchemaVersion>
        <Destination>
          <S3BucketDestination>
            <Format>CSV</Format>
            <BucketAccountId>123456789012</BucketAccountId>
            <Bucket>arn:aws:s3:::destination-bucket</Bucket>
            <Prefix>destination-prefix</Prefix>
          </S3BucketDestination>
        </Destination>
      </DataExport>
    </StorageClassAnalysis>
  </AnalyticsConfiguration>
  <AnalyticsConfiguration>
    <Id>report1</Id>
    <Filter>
      <And>
        <Prefix>images/</Prefix>
```

```
<Tag>
  <Key>dog</Key>
  <Value>bulldog</Value>
</Tag>
</And>
</Filter>
<StorageClassAnalysis>
  <DataExport>
    <OutputSchemaVersion>V_1</OutputSchemaVersion>
    <Destination>
      <S3BucketDestination>
        <Format>CSV</Format>
        <BucketAccountId>123456789012</BucketAccountId>
        <Bucket>arn:aws:s3:::destination-bucket</Bucket>
        <Prefix>destination-prefix</Prefix>
      </S3BucketDestination>
    </Destination>
  </DataExport>
</StorageClassAnalysis>
</AnalyticsConfiguration>
...
<IsTruncated>false</IsTruncated>
<!-- If ContinuationToken was provided in the request. -->
<ContinuationToken>...</ContinuationToken>
<!-- if IsTruncated == true -->
<IsTruncated>true</IsTruncated>
<NextContinuationToken>...</NextContinuationToken>
</ListBucketAnalyticsConfigurationResult>
```

For an example of using the ContinuationToken with a list, see [Example 4: Using a Continuation Token \(p. 109\)](#).

## Related Resources

- [GET Bucket analytics \(p. 126\)](#)
- [DELETE Bucket analytics \(p. 80\)](#)
- [PUT Bucket analytics \(p. 242\)](#)

# List Bucket Inventory Configurations

## Description

This implementation of the `GET` operation returns a list of inventory configurations for the bucket. You can have up to 1,000 analytics configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. Always check the `IsTruncated` element in the response. If there are no more configurations to list, `IsTruncated` is set to false. If there are more configurations to list, `IsTruncated` is set to true, and there is a value in `NextContinuationToken`. You use the `NextContinuationToken` value to continue the pagination of the list by passing the value in `continuation-token` in the request to `GET` the next page.

To use this operation, you must have permissions to perform the `s3:GetInventoryConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about Amazon S3 inventory feature, see [Amazon S3 Inventory](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?inventory HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of `GET` uses the parameters in the following table.

Parameter	Description	Required
<code>continuation-token</code>	When the Amazon S3 response to this API call is truncated (that is, when the <code>IsTruncated</code> response element value is true), the response also includes the <code>NextContinuationToken</code> element. You can use the value of this element in the next request as the <code>continuation-token</code> to list the next page. The continuation token is an opaque value that Amazon S3 understands. Type: String  Default: None	No

### Request Elements

This implementation of the operation does not use request elements.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
ContinuationToken	The marker that is used as a starting point for this inventory configuration list response. This value is present if it was sent in the request.  Type: String  Ancestor: <code>ListInventoryConfigurationsResult</code>
IsTruncated	Tells whether the returned list of inventory configurations is complete. A value of true indicates that the list is not complete and the <code>NextContinuationToken</code> is provided for a subsequent request.  Type: Boolean  Ancestor: <code>ListInventoryConfigurationsResult</code>
InventoryConfiguration	Contains the inventory configuration. For the XML structure, see <a href="#">GET Bucket Inventory (p. 139)</a> .  Type: Container  Ancestor: <code>ListInventoryConfigurationsResult</code>
ListInventoryConfigurationsResult	The list of inventory configurations for a bucket.  Type: Container
NextContinuationToken	The marker that is used to continue an inventory configuration listing that has been truncated. Use the <code>NextContinuationToken</code> from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.  Type: String  Ancestor: <code>ListInventoryConfigurationsResult</code>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Listing Inventory Configurations

The following request returns the inventory configurations in example-bucket.

#### Sample Request

```
GET /?inventory HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: 20160430T233541Z
Authorization: authorization string
Content-Type: text/plain
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXR3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ListInventoryConfigurationsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <InventoryConfiguration>
        <Id>report1</Id>
        <IsEnabled>true</IsEnabled>
        <Destination>
            <S3BucketDestination>
                <Format>CSV</Format>
                <AccountId>123456789012</AccountId>
                <Bucket>arn:aws:s3:::destination-bucket</Bucket>
                <Prefix>prefix1</Prefix>
            </S3BucketDestination>
        </Destination>
        <Schedule>
            <Frequency>Daily</Frequency>
        </Schedule>
        <Filter>
            <Prefix>prefix/One</Prefix>
        </Filter>
        <IncludedObjectVersions>All</IncludedObjectVersions>
        <OptionalFields>
            <Field>Size</Field>
            <Field>LastModifiedDate</Field>
            <Field>ETag</Field>
            <Field>StorageClass</Field>
            <Field>IsMultipartUploaded</Field>
            <Field>ReplicationStatus</Field>
        </OptionalFields>
    </InventoryConfiguration>
    <InventoryConfiguration>
        <Id>report2</Id>
```

```
<Enabled>true</Enabled>
<Destination>
    <S3BucketDestination>
        <Format>CSV</Format>
        <AccountId>123456789012</AccountId>
        <Bucket>arn:aws:s3:::bucket2</Bucket>
        <Prefix>prefix2</Prefix>
    </S3BucketDestination>
</Destination>
<Schedule>
    <Frequency>Daily</Frequency>
</Schedule>
<Filter>
    <Prefix>prefix/Two</Prefix>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
<OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>ReplicationStatus</Field>
    <Field>ObjectLockRetainUntilDate</Field>
    <Field>ObjectLockMode</Field>
    <Field>ObjectLockLegalHoldStatus</Field>
</OptionalFields>
</InventoryConfiguration>
<InventoryConfiguration>
    <Id>report3</Id>
    <Enabled>true</Enabled>
    <Destination>
        <S3BucketDestination>
            <Format>CSV</Format>
            <AccountId>123456789012</AccountId>
            <Bucket>arn:aws:s3:::bucket3</Bucket>
            <Prefix>prefix3</Prefix>
        </S3BucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
        <Prefix>prefix/Three</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
        <Field>Size</Field>
        <Field>LastModifiedDate</Field>
        <Field>ETag</Field>
        <Field>StorageClass</Field>
        <Field>IsMultipartUploaded</Field>
        <Field>ReplicationStatus</Field>
    </OptionalFields>
</InventoryConfiguration>
...
<IsTruncated>false</IsTruncated>
<!-- If ContinuationToken was provided in the request. -->
<ContinuationToken>...</ContinuationToken>
<!-- if IsTruncated == true -->
<IsTruncated>true</IsTruncated>
<NextContinuationToken>...</NextContinuationToken>
</ListBucketAnalyticsConfigurationResult>
</ListInventoryConfigurationsResult>
```

For an example of using the ContinuationToken with a list, see [Example 4: Using a Continuation Token \(p. 109\)](#).

## Related Resources

- [GET Bucket Inventory \(p. 139\)](#)
- [DELETE Bucket inventory \(p. 86\)](#)
- [PUT Bucket inventory \(p. 258\)](#)

# List Bucket Metrics Configurations

## Description

This implementation of the GET operation returns a list of Amazon CloudWatch metrics configurations for the bucket. The metrics configurations are only for the request metrics of the bucket and do not provide information on daily storage metrics. You can have up to 1,000 configurations per bucket.

This operation supports list pagination and does not return more than 100 configurations at a time. Always check the `IsTruncated` element in the response. If there are no more configurations to list, `IsTruncated` is set to false. If there are more configurations to list, `IsTruncated` is set to true, and there is a value in `NextContinuationToken`. You use the `NextContinuationToken` value to continue the pagination of the list by passing the value in `continuation-token` in the request to GET the next page.

To use this operation, you must have permissions to perform the `s3:GetMetricsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information about metrics configurations and CloudWatch request metrics, see [Monitoring Metrics with Amazon CloudWatch](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?metrics HTTP/1.1
HOST: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

Parameter	Description	Required
<code>continuation-token</code>	When the Amazon S3 response to this API call is truncated (that is, when the <code>IsTruncated</code> response element value is true), the response also includes the <code>NextContinuationToken</code> element. You can use the value of that element in the next request as the <code>continuation-token</code> to list the next page. The continuation token is an opaque value that Amazon S3 understands. Type: String Default: None	No

### Request Headers

This operation uses only Request Headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

This operation does not use request elements.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
IsTruncated	Tells whether the returned list of metrics configurations is complete. A value of <code>true</code> indicates that the list is not complete, and the <code>NextContinuationToken</code> is provided for a subsequent request.  Type: Boolean  Ancestor: <code>ListMetricsConfigurationResult</code>
ContinuationToken	The marker that is used as a starting point for this metrics configuration list response. This value is present if it was sent in the request.  Type: String  Ancestor: <code>ListMetricsConfigurationResult</code>
NextContinuationToken	The marker used to continue a metrics configuration listing that has been truncated. Use the <code>NextContinuationToken</code> from a previously truncated list response to continue the listing. The continuation token is an opaque value that Amazon S3 understands.  Type: String  Ancestor: <code>ListMetricsConfigurationResult</code>
<code>ListMetricsConfigurationsResult</code>	The list of metrics configurations for a bucket.  Type: Container

## Examples

### Sample Request

```
GET /?metrics HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 758

<?xml version="1.0" encoding="UTF-8"?>
<ListMetricsConfigurationsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <MetricsConfiguration>
        <Id>EntireBucket</Id>
    </MetricsConfiguration>
    <MetricsConfiguration>
        <Id>Documents</Id>
        <Filter>
            <Prefix>documents/</Prefix>
        </Filter>
    </MetricsConfiguration>
    <MetricsConfiguration>
        <Id>BlueDocuments</Id>
        <Filter>
            <And>
                <Prefix>documents/</Prefix>
                <Tag>
                    <Key>class</Key>
                    <Value>blue</Value>
                </Tag>
            </And>
        </Filter>
    </MetricsConfiguration>
    <IsTruncated>false</IsTruncated>
</ListMetricsConfigurationsResult>
```

## Related Resources

- [PUT Bucket metrics \(p. 285\)](#)
- [DELETE Bucket metrics \(p. 90\)](#)
- [GET Bucket metrics \(p. 160\)](#)
- [Monitoring Metrics with Amazon CloudWatch](#) in the *Amazon Simple Storage Service Developer Guide*.

# List Multipart Uploads

## Description

This operation lists in-progress multipart uploads. An in-progress multipart upload is a multipart upload that has been initiated using the Initiate Multipart Upload request, but has not yet been completed or aborted.

This operation returns at most 1,000 multipart uploads in the response. 1,000 multipart uploads is the maximum number of uploads a response can include, which is also the default value. You can further limit the number of uploads in a response by specifying the `max-uploads` parameter in the response. If additional multipart uploads satisfy the list criteria, the response will contain an `IsTruncated` element with the value `true`. To list the additional multipart uploads, use the `key-marker` and `upload-id-marker` request parameters.

In the response, the uploads are sorted by key. If your application has initiated more than one multipart upload using the same object key, then uploads in the response are first sorted by key. Additionally, uploads are sorted in ascending order within each key by the upload initiation time.

For more information on multipart uploads, see [Uploading Objects Using Multipart Upload](#) in the *Amazon Simple Storage Service Developer Guide*.

For information on permissions required to use the multipart upload API, see [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string
```

### Request Parameters

Parameter	Description	Required
<code>delimiter</code>	<p>Character you use to group keys.</p> <p>All keys that contain the same string between the <code>prefix</code>, if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, <code>CommonPrefixes</code>. If you don't specify the <code>prefix</code> parameter, then the substring starts at the beginning of the key. The keys that are grouped under <code>CommonPrefixes</code> result element are not returned elsewhere in the response.</p> <p>Type: String</p>	No
<code>encoding-type</code>	<p>Requests Amazon S3 to encode the response and specifies the encoding method to use.</p> <p>An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII</p>	No

Parameter	Description	Required
	<p>value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid value: url</p>	
max-uploads	<p>Sets the maximum number of multipart uploads, from 1 to 1,000, to return in the response body. 1,000 is the maximum number of uploads that can be returned in a response.</p> <p>Type: Integer</p> <p>Default: 1,000</p>	No
key-marker	<p>Together with upload-id-marker, this parameter specifies the multipart upload after which listing should begin.</p> <p>If upload-id-marker is not specified, only the keys lexicographically greater than the specified key-marker will be included in the list.</p> <p>If upload-id-marker is specified, any multipart uploads for a key equal to the key-marker might also be included, provided those multipart uploads have upload IDs lexicographically greater than the specified upload-id-marker.</p> <p>Type: String</p>	No
prefix	<p>Lists in-progress uploads only for those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different grouping of keys. (You can think of using prefix to make groups in the same way you'd use a folder in a file system.)</p> <p>Type: String</p>	No
upload-id-marker	<p>Together with key-marker, specifies the multipart upload after which listing should begin. If key-marker is not specified, the upload-id-marker parameter is ignored. Otherwise, any multipart uploads for a key equal to the key-marker might be included in the list only if they have an upload ID lexicographically greater than the specified upload-id-marker.</p> <p>Type: String</p>	No

## Request Headers

This operation uses only Request Headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

This operation does not use request elements.

# Responses

## Response Headers

This operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

Name	Description
ListMultipartUploadsResult	<p>Container for the response.</p> <p>Children: Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MaxUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated</p> <p>Type: Container</p> <p>Ancestor: None</p>
Bucket	<p>Name of the bucket to which the multipart upload was initiated.</p> <p>Type: String</p> <p>Ancestor: ListMultipartUploadsResult</p>
KeyMarker	<p>The key at or after which the listing began.</p> <p>Type: String</p> <p>Ancestor: ListMultipartUploadsResult</p>
UploadIdMarker	<p>Upload ID after which listing began.</p> <p>Type: String</p> <p>Ancestor: ListMultipartUploadsResult</p>
NextKeyMarker	<p>When a list is truncated, this element specifies the value that should be used for the key-marker request parameter in a subsequent request.</p> <p>Type: String</p> <p>Ancestor: ListMultipartUploadsResult</p>
NextUploadIdMarker	<p>When a list is truncated, this element specifies the value that should be used for the upload-id-marker request parameter in a subsequent request.</p> <p>Type: String</p> <p>Ancestor: ListMultipartUploadsResult</p>
Encoding-Type	Encoding type used by Amazon S3 to encode object key names in the XML response.

Name	Description
	<p>If you specify encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the following response elements:</p> <p><code>Delimiter</code>, <code>KeyMarker</code>, <code>Prefix</code>, <code>NextKeyMarker</code>, <code>Key</code>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
<code>MaxUploads</code>	<p>Maximum number of multipart uploads that could have been included in the response.</p> <p>Type: Integer</p> <p>Ancestor: <code>ListMultipartUploadsResult</code></p>
<code>IsTruncated</code>	<p>Indicates whether the returned list of multipart uploads is truncated. A value of <code>true</code> indicates that the list was truncated. The list can be truncated if the number of multipart uploads exceeds the limit allowed or specified by <code>MaxUploads</code>.</p> <p>Type: Boolean</p> <p>Ancestor: <code>ListMultipartUploadsResult</code></p>
<code>Upload</code>	<p>Container for elements related to a particular multipart upload. A response can contain zero or more <code>Upload</code> elements.</p> <p>Type: Container</p> <p>Children: <code>Key</code>, <code>UploadId</code>, <code>InitiatorOwner</code>, <code>StorageClass</code>, <code>Initiated</code></p> <p>Ancestor: <code>ListMultipartUploadsResult</code></p>
<code>Key</code>	<p>Key of the object for which the multipart upload was initiated.</p> <p>Type: Integer</p> <p>Ancestor: <code>Upload</code></p>
<code>UploadId</code>	<p>Upload ID that identifies the multipart upload.</p> <p>Type: Integer</p> <p>Ancestor: <code>Upload</code></p>

Name	Description
<b>Initiator</b>	<p>Container element that identifies who initiated the multipart upload. If the initiator is an AWS account, this element provides the same information as the <code>Owner</code> element. If the initiator is an IAM User, then this element provides the user ARN and display name.</p> <p>Children: <code>ID</code>, <code>DisplayName</code></p> <p>Type: Container</p> <p>Ancestor: <code>Upload</code></p>
<b>ID</b>	<p>If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.</p> <p>Type: String</p> <p>Ancestor: <code>Initiator</code>, <code>Owner</code></p>
<b>DisplayName</b>	<p>Principal's name.</p> <p>Type: String</p> <p>Ancestor: <code>Initiator</code> , <code>Owner</code></p>
<b>Owner</b>	<p>Container element that identifies the object owner, after the object is created. If multipart upload is initiated by an IAM user, this element provides a the parent account ID and display name.</p> <p>Type: Container</p> <p>Children: <code>ID</code>, <code>DisplayName</code></p> <p>Ancestor: <code>Upload</code></p>
<b>StorageClass</b>	<p>The class of storage (<code>STANDARD</code> or <code>REDUCED_REDUNDANCY</code>) that will be used to store the object when the multipart upload is complete.</p> <p>Type: String</p> <p>Ancestor: <code>Upload</code></p>
<b>Initiated</b>	<p>Date and time at which the multipart upload was initiated.</p> <p>Type: Date</p> <p>Ancestor: <code>Upload</code></p>
<b>ListMultipartUploadsResult.Prefix</b>	<p>When a prefix is provided in the request, this field contains the specified prefix. The result contains only keys starting with the specified prefix.</p> <p>Type: String</p> <p>Ancestor: <code>ListMultipartUploadsResult</code></p>

Name	Description
Delimiter	<p>Contains the delimiter you specified in the request. If you don't specify a delimiter in your request, this element is absent from the response.</p> <p>Type: String</p> <p>Ancestor: ListMultipartUploadsResult</p>
CommonPrefixes	<p>If you specify a delimiter in the request, then the result returns each distinct key prefix containing the delimiter in a CommonPrefixes element. The distinct key prefixes are returned in the Prefix child element.</p> <p>Type: Container</p> <p>Ancestor: ListMultipartUploadsResult</p>
CommonPrefixes.Prefix	<p>If the request does not include the Prefix parameter, then this element shows only the substring of the key that precedes the first occurrence of the delimiter character. These keys are not returned anywhere else in the response.</p> <p>If the request includes the Prefix parameter, then this element shows the substring of the key from the beginning to the first occurrence of the delimiter after the prefix.</p> <p>Type: String</p> <p>Ancestor: CommonPrefixes</p>

## Examples

### Sample Request

The following request lists three multipart uploads. The request specifies the `max-uploads` request parameter to set the maximum number of multipart uploads to return in the response body.

```
GET /?uploads&max-uploads=3 HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

### Sample Response

The following sample response indicates that the multipart upload list was truncated and provides the `NextKeyMarker` and the `NextUploadIdMarker` elements. You specify these values in your subsequent requests to read the next set of multipart uploads. That is, send a subsequent request specifying `key-marker=my-movie2.m2ts` (value of the `NextKeyMarker` element) and `upload-id-marker=YW55IGlkZWEgd2h5IGVsdlmuZydzIHVwbG9hZCBmYWlsZWQ` (value of the `NextUploadIdMarker`).

The sample response also shows a case of two multipart uploads in progress with the same key (`my-movie.m2ts`). That is, the response shows two uploads with the same key. This response shows the uploads sorted by key, and within each key the uploads are sorted in ascending order by the time the multipart upload was initiated.

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOfg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 1330
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<ListMultipartUploadsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>bucket</Bucket>
  <KeyMarker></KeyMarker>
  <UploadIdMarker></UploadIdMarker>
  <NextKeyMarker>my-movie.m2ts</NextKeyMarker>
  <NextUploadIdMarker>YW55IGlkZWEgd2h5IGVsdluZydzIHVwbG9hZCBmYWlsZWQ</NextUploadIdMarker>
  <MaxUploads>3</MaxUploads>
  <IsTruncated>true</IsTruncated>
  <Upload>
    <Key>my-divisor</Key>
    <UploadId>XMgbGlrZSBlbHZpbmcnycBub3QgaGF2aW5nIG11Y2ggbHVjaw</UploadId>
    <Initiator>
      <ID>arn:aws:iam::111122223333:user/user1-11111a31-17b5-4fb7-9df5-b11111f13de</ID>
      <DisplayName>user1-11111a31-17b5-4fb7-9df5-b11111f13de</DisplayName>
    </Initiator>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>OwnerDisplayName</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:48:33.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>my-movie.m2ts</Key>
    <UploadId>VXBsb2FkIE1EIGZvcIBlbHZpbmcnycBteS1tb3ZpZS5tMnRzIHVwbG9hZA</UploadId>
    <Initiator>
      <ID>b1d16700c70b0b05597d7acd6a3f92be</ID>
      <DisplayName>InitiatorDisplayName</DisplayName>
    </Initiator>
    <Owner>
      <ID>b1d16700c70b0b05597d7acd6a3f92be</ID>
      <DisplayName>OwnerDisplayName</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:48:33.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>my-movie.m2ts</Key>
    <UploadId>YW55IGlkZWEgd2h5IGVsdluZydzIHVwbG9hZCBmYWlsZWQ</UploadId>
    <Initiator>
      <ID>arn:aws:iam::444455556666:user/user1-22222a31-17b5-4fb7-9df5-b222222f13de</ID>
      <DisplayName>user1-22222a31-17b5-4fb7-9df5-b222222f13de</DisplayName>
    </Initiator>
    <Owner>
      <ID>b1d16700c70b0b05597d7acd6a3f92be</ID>
      <DisplayName>OwnerDisplayName</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
    <Initiated>2010-11-10T20:49:33.000Z</Initiated>
  </Upload>
</ListMultipartUploadsResult>
```

## Sample Request Using the Delimiter and the Prefix Parameters

Assume you have a multipart upload in progress for the following keys in your bucket, example-bucket.

```
photos/2006/January/sample.jpg
photos/2006/February/sample.jpg
photos/2006/March/sample.jpg
videos/2006/March/sample.wmv
sample.jpg
```

The following list multipart upload request specifies the delimiter parameter with value "/".

```
GET /?uploads&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

The following sample response lists multipart uploads on the specified bucket, example-bucket.

The response returns multipart upload for the sample.jpg key in an <Upload> element.

However, because all the other keys contain the specified delimiter, a distinct substring, from the beginning of the key to the first occurrence of the delimiter, from each of these keys is returned in a <CommonPrefixes> element. The key substrings, photos/ and videos/, in the <CommonPrefixes> element indicate that there are one or more in-progress multipart uploads with these key prefixes.

This is a useful scenario if you use key prefixes for your objects to create a logical folder like structure. In this case you can interpret the result as the folders photos/ and videos/ have one or more multipart uploads in progress.

```
<ListMultipartUploadsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Bucket>example-bucket</Bucket>
<KeyMarker/>
<UploadIdMarker/>
<NextKeyMarker>sample.jpg</NextKeyMarker>

<NextUploadIdMarker>Xgw4MJT6ZPAVxpYOSAuGN7q4uWJJM22ZYg1W99trdp4tp088.PT6.MhO0w2E17eutfAvQfQWoajgE_W2gp
</NextUploadIdMarker>
<Delimiter></Delimiter>
<Prefix/>
<MaxUploads>1000</MaxUploads>
<IsTruncated>false</IsTruncated>
<Upload>
  <Key>sample.jpg</Key>

  <UploadId>Agw4MJT6ZPAVxpYOSAuGN7q4uWJJM22ZYg1N99trdp4tp088.PT6.MhO0w2E17eutfAvQfQWoajgE_W2gp
  </UploadId>
  <Initiator>
    <ID>314133b66967d86f031c7249d1d9a80249109428335cd0ef1cdc487b4566cb1b</ID>
    <DisplayName>s3-nickname</DisplayName>
  </Initiator>
  <Owner>
    <ID>314133b66967d86f031c7249d1d9a80249109428335cd0ef1cdc487b4566cb1b</ID>
    <DisplayName>s3-nickname</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
```

```
<Initiated>2010-11-26T19:24:17.000Z</Initiated>
</Upload>
<CommonPrefixes>
  <Prefix>photos/</Prefix>
</CommonPrefixes>
<CommonPrefixes>
  <Prefix>videos/</Prefix>
</CommonPrefixes>
</ListMultipartUploadsResult>
```

In addition to the delimiter parameter you can filter results by adding a `prefix` parameter as shown in the following request.

```
GET /uploads&delimiter=/&prefix=photos/2006/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

In this case the response will include only multipart uploads for keys that start with the specified prefix. The value returned in the `<CommonPrefixes>` element is a substring from the beginning of the key to the first occurrence of the specified delimiter after the prefix.

```
<?xml version="1.0" encoding="UTF-8"?>
<ListMultipartUploadsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <KeyMarker/>
  <UploadIdMarker/>
  <NextKeyMarker/>
  <NextUploadIdMarker/>
  <Delimiter>/</Delimiter>
  <Prefix>photos/2006/</Prefix>
  <MaxUploads>1000</MaxUploads>
  <IsTruncated>false</IsTruncated>
  <CommonPrefixes>
    <Prefix>photos/2006/February/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/January/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/March/</Prefix>
  </CommonPrefixes>
</ListMultipartUploadsResult>
```

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)
- [Upload Part \(p. 508\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Parts \(p. 502\)](#)

# PUT Bucket

## Description

This implementation of the `PUT` operation creates a new bucket. To create a bucket, you must register with Amazon S3 and have a valid AWS Access Key ID to authenticate requests. Anonymous requests are never allowed to create buckets. By creating the bucket, you become the bucket owner.

Not every string is an acceptable bucket name. For information on bucket naming restrictions, see [Working with Amazon S3 Buckets](#).

By default, the bucket is created in the US East (N. Virginia) region. You can optionally specify a region in the request body. You might choose a region to optimize latency, minimize costs, or address regulatory requirements. For example, if you reside in Europe, you will probably find it advantageous to create buckets in the EU (Ireland) region. For more information, see [How to Select a Region for Your Buckets](#).

### Note

If you create a bucket in a region other than US East (N. Virginia) region, your application must be able to handle 307 redirect. For more information, go to [Virtual Hosting of Buckets in Amazon Simple Storage Service Developer Guide](#).

When creating a bucket using this operation, you can optionally specify the accounts or groups that should be granted specific permissions on the bucket. There are two ways to grant the appropriate permissions using the request headers.

- Specify a canned ACL using the `x-amz-acl` request header. For more information, see [Canned ACL](#) in the *Amazon Simple Storage Service Developer Guide*.
- Specify access permissions explicitly using the `x-amz-grant-read`, `x-amz-grant-write`, `x-amz-grant-read-acp`, `x-amz-grant-write-acp`, `x-amz-grant-full-control` headers. These headers map to the set of permissions Amazon S3 supports in an ACL. For more information, go to [Access Control List \(ACL\) Overview](#) in the *Amazon Simple Storage Service Developer Guide*.

### Note

You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

## Requests

### Syntax

```
PUT / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
<CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LocationConstraint>BucketRegion</LocationConstraint>
</CreateBucketConfiguration>
```

### Note

The syntax shows some of the request headers. For a complete list, see the Request Headers section.

### Note

If you send your create bucket request to the `s3.amazonaws.com` endpoint, the request goes to the `us-east-1` region. Accordingly, the signature calculations in Signature Version 4 must

use `us-east-1` as region, even if the location constraint in the request specifies another region where the bucket is to be created.

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

When creating a bucket, you can grant permissions to individual AWS accounts or predefined groups defined by Amazon S3. This results in creation of the Access Control List (ACL) on the bucket. For more information, see [Using ACLs](#). You have the following two ways to grant these permissions:

- **Specify a canned ACL** — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, go to [Canned ACL](#).

Name	Description	Required
<code>x-amz-acl</code>	<p>The canned ACL to apply to the bucket you are creating. For more information, go to <a href="#">Canned ACL</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Valid Values: <code>private</code>   <code>public-read</code>   <code>public-read-write</code>   <code>aws-exec-read</code>   <code>authenticated-read</code>   <code>bucket-owner-read</code>   <code>bucket-owner-full-control</code></p>	No

- **Specify access permissions explicitly** — If you want to explicitly grant access permissions to specific AWS accounts or groups, you use the following headers. Each of these headers maps to specific permissions Amazon S3 supports in an ACL. For more information, go to [Access Control List \(ACL\) Overview](#). In the header value, you specify a list of grantees who get the specific permission

Name	Description	Required
<code>x-amz-grant-read</code>	<p>Allows grantee to list the objects in the bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
<code>x-amz-grant-write</code>	<p>Allows grantee to create, overwrite, and delete any object in the bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No

Name	Description	Required
x-amz-grant-read-acp	Allows grantee to read the bucket ACL. Type: String Default: None Constraints: None	No
x-amz-grant-write-acp	Allows grantee to write the ACL for the applicable bucket. Type: String Default: None Constraints: None	No
x-amz-grant-full-control	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket. Type: String Default: None Constraints: None	No

You specify each grantee as a type=value pair, where the type can be one of the following::

- **emailAddress** — if value specified is the email address of an AWS account
- **id** — if value specified is the canonical user ID of an AWS account
- **uri** — if granting permission to a predefined group.

For example, the following x-amz-grant-read header grants list objects permission to the AWS accounts identified by their email addresses.

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

For more information see, [ACL Overview](#).

## Request Elements

Name	Description	Required
CreateBucketConfiguration	Container for bucket configuration settings. Type: Container Ancestor: None	No
LocationConstraint	Specifies the region where the bucket will be created. If you are creating a bucket on the US East (N. Virginia) region (us-east-1), you do not need to specify the location constraint. Type: Enum	No

Name	Description	Required
	<p>Valid Values: For a list of all the Amazon S3 supported location constraints by region, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p> <p>Default: US East (N. Virginia) region</p> <p>Ancestor: CreateBucketConfiguration</p>	

## Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request creates a bucket named `colorpictures`.

```
PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 0
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

### Sample Request: Setting the region of a bucket

The following request sets the region the bucket to EU.

```
PUT / HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124
```

```
<CreateBucketConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <LocationConstraint>EU</LocationConstraint>
</CreateBucketConfiguration >
```

## Sample Response

### Sample Request: Creating a bucket and configuring access permission using a canned ACL

This request creates a bucket named "colorpictures" and sets the ACL to private.

```
PUT / HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 0
x-amz-acl: private
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

### Sample Request: Creating a bucket and configuring access permissions explicitly

This request creates a bucket named colorpictures and grants WRITE permission to the AWS account identified by an email address.

```
PUT HTTP/1.1
Host: colorpictures.s3.amazonaws.com
x-amz-date: Sat, 07 Apr 2012 00:54:40 GMT
Authorization: authorization string
x-amz-grant-write: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

## Sample Response

```
HTTP/1.1 200 OK
```

## Related Resources

- [PUT Object \(p. 412\)](#)
- [DELETE Bucket \(p. 78\)](#)

# PUT Bucket accelerate

## Description

This implementation of the `PUT` operation uses the `accelerate` subresource to set the Transfer Acceleration state of an existing bucket. Amazon S3 Transfer Acceleration is a bucket-level feature that enables you to perform faster data transfers to Amazon S3.

To use this operation, you must have permission to perform the `s3:PutAccelerateConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

The Transfer Acceleration state of a bucket can be set to one of the following two values:

- **Enabled** – Enables accelerated data transfers to the bucket.
- **Suspended** – Disables accelerated data transfers to the bucket.

The [GET Bucket accelerate \(p. 120\)](#) operation returns the transfer acceleration state of a bucket.

After setting the Transfer Acceleration state of a bucket to `Enabled`, it might take up to thirty minutes before the data transfer rates to the bucket increase.

The name of the bucket used for Transfer Acceleration must be DNS-compliant and must not contain periods (".").

For more information about transfer acceleration, see [Transfer Acceleration](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?accelerate HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Transfer acceleration configuration in the request body
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Body

In the request, you specify the acceleration configuration in the request body. The acceleration configuration is specified as XML. The following is an example of an acceleration configuration used in a request. The Status indicates whether to set the transfer acceleration state to Enabled or Suspended.

```
<AccelerateConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>transfer acceleration state</Status>
</AccelerateConfiguration>
```

The following table describes the XML elements in the acceleration configuration:

Name	Description	Required
AccelerateConfiguration	Container for setting the transfer acceleration state.  Type: Container  Children: Status  Ancestor: None	Yes
Status	Sets the transfer acceleration state of the bucket.  Type: Enum  Valid Values: Enabled   Suspended  Ancestor: AccelerateConfiguration	Yes

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Add Transfer Acceleration Configuration to Set Acceleration Status

The following is an example of a `PUT /?accelerate` request that enables transfer acceleration for the bucket named `examplebucket`.

```
PUT /?accelerate HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: length

<AccelerateConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</AccelerateConfiguration>
```

The following is an example response:

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 11 Apr 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [GET Bucket accelerate \(p. 120\)](#)
- [PUT Bucket \(p. 227\)](#)

# PUT Bucket acl

## Description

This implementation of the `PUT` operation uses the `acl` subresource to set the permissions on an existing bucket using access control lists (ACL). For more information, go to [Using ACLs](#). To set the ACL of a bucket, you must have `WRITE_ACP` permission.

You can use one of the following two ways to set a bucket's permissions:

- Specify the ACL in the request body
- Specify permissions using request headers

### Note

You cannot specify access permission using both the body and the request headers.

Depending on your application needs, you may choose to set the ACL on a bucket using either the request body or the headers. For example, if you have an existing application that updates a bucket ACL using the request body, then you can continue to use that approach.

## Requests

### Syntax

The following request shows the syntax for sending the ACL in the request body. If you want to use headers to specify the permissions for the bucket, you cannot send the ACL in the request body. Instead, see Request Headers section for a list of headers you can use.

```
PUT /?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))

<AccessControlPolicy>
  <Owner>
    <ID>ID</ID>
    <DisplayName>EmailAddress</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>ID</ID>
        <DisplayName>EmailAddress</DisplayName>
      </Grantee>
      <Permission>Permission</Permission>
    </Grant>
    ...
  </AccessControlList>
</AccessControlPolicy>
```

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

You can use the following request headers in addition to the [Common Request Headers \(p. 2\)](#).

These headers enable you to set access permissions using one of the following methods:

- Specify a canned ACL, or
- Specify the permission for each grantee explicitly

Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see [Canned ACL](#). To grant access permissions by specifying canned ACLs, you use the following header and specify the canned ACL name as its value. If you use this header, you cannot use other access control specific headers in your request.

Name	Description	Required
x-amz-acl	<p>Sets the ACL of the bucket using the specified canned ACL. For more information, go to <a href="#">Canned ACL</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Valid Values: private   public-read   public-read-write   authenticated-read</p> <p>Default: private</p>	No

If you need to grant individualized access permissions on a bucket, you can use the following "x-amz-grant-permission" headers. When using these headers you specify explicit access permissions and grantees (AWS accounts or a Amazon S3 groups) who will receive the permission. If you use these ACL specific headers, you cannot use x-amz-acl header to set a canned ACL.

**Note**

Each of the following request headers maps to specific permissions Amazon S3 supports in an ACL. For more information go to [Access Control List \(ACL\) Overview](#).

Name	Description	Required
x-amz-grant-read	<p>Allows the specified grantee(s) to list the objects in the bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-write	<p>Allows the specified grantee(s) to create, overwrite, and delete any object in the bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-read-acp	Allows the specified grantee(s) to read the bucket ACL.	No

Name	Description	Required
	Type: String Default: None Constraints: None	
x-amz-grant-write-acp	Allows the specified grantee(s) to write the ACL for the applicable bucket.  Type: String Default: None Constraints: None	No
x-amz-grant-full-control	Allows the specified grantee(s) the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket.  Type: String Default: None Constraints: None	No

For each of these headers, the value is a comma-separated list of one or more grantees. You specify each grantee as a type=value pair, where the type can be one of the following:

- **emailAddress** — if value specified is the email address of an AWS account
- **id** — if value specified is the canonical User ID of an AWS account
- **uri** — if granting permission to a predefined Amazon S3 group.

For example, the following x-amz-grant-write header grants create, overwrite, and delete objects permission to LogDelivery group predefined by Amazon S3 and two AWS accounts identified by their email addresses.

```
x-amz-grant-write: uri="http://acs.amazonaws.com/groups/s3/LogDelivery",
  emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

For more information, go to [Access Control List \(ACL\) Overview](#). For more information about bucket logging, go to [Server Access Logging](#).

## Request Elements

If you decide to use the request body to specify an ACL, you must use the following elements.

**Note**

If you request the request body, you cannot use the request headers to set an ACL.

Name	Description	Required
AccessControlList	Container for Grant, Grantee, and Permission  Type: Container  Ancestors: AccessControlPolicy	No

Name	Description	Required
AccessControlPolicy	<p>Contains the elements that set the ACL permissions for an object per grantee.</p> <p>Type: String</p> <p>Ancestors: None</p>	No
DisplayName	<p>Screen name of the bucket owner.</p> <p>Type: String</p> <p>Ancestors: AccessControlPolicy.Owner</p>	No
Grant	<p>Container for the grantee and his or her permissions.</p> <p>Type: Container</p> <p>Ancestors: AccessControlPolicy.AccessControlList</p>	No
Grantee	<p>The subject whose permissions are being set. For more information, see <a href="#">Grantee Values (p. 238)</a>.</p> <p>Type: String</p> <p>Ancestors: AccessControlPolicy.AccessControlList.Grant</p>	No
ID	<p>ID of the bucket owner, or the ID of the grantee.</p> <p>Type: String</p> <p>Ancestors: AccessControlPolicy.Owner   AccessControlPolicy.AccessControlList.Grant</p>	No
Owner	<p>Container for the bucket owner's display name and ID.</p> <p>Type: Container</p> <p>Ancestors: AccessControlPolicy</p>	Yes
Permission	<p>Specifies the permission given to the grantee.</p> <p>Type: String</p> <p>Valid Values: FULL_CONTROL   WRITE   WRITE_ACP   READ   READ_ACP</p> <p>Ancestors: AccessControlPolicy.AccessControlList.Grant</p>	No

## Grantee Values

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By the person's ID:

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

DisplayName is optional and ignored in the request.

- By Email address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="AmazonCustomerByEmail"><EmailAddress><replaceable>Grantees@email.com</
replaceable></EmailAddress></Grantee>
```

The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By URI:

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

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not return response elements.

### Special Errors

This operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request: Access permissions specified in the body

The following request grants access permission to the existing examplebucket bucket. The request specifies the ACL in the body. In addition to granting full control to the bucket owner, the XML specifies the following grants.

- Grant AllUsers group READ permission on the bucket.
- Grant the LogDelivery group WRITE permission on the bucket.
- Grant an AWS account, identified by email address, WRITE\_ACP permission.
- Grant an AWS account, identified by canonical user ID, READ\_ACP permission.

```
PUT ?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```

```
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2012 20:04:21 GMT
Authorization: authorization string

<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID</ID>
    <DisplayName>OwnerDisplayName</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID</ID>
        <DisplayName>OwnerDisplayName</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission xmlns="">READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
      </Grantee>
      <Permission xmlns="">WRITE</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AmazonCustomerByEmail">
        <EmailAddress xmlns="">xyz@amazon.com</EmailAddress>
      </Grantee>
      <Permission xmlns="">WRITE_ACP</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID xmlns="">f30716ab7115dcba4a5ef76e9d74b8e20567f63TestAccountCanonicalUserID</ID>
      </Grantee>
      <Permission xmlns="">READ_ACP</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: NxqO3PNiMHXXGwjgv15LLgUoAmPVmG0xtZw2sxePXLhpIvcyouXDrcQUaWWXcOK0
x-amz-request-id: C651BC9B4E1BD401
Date: Thu, 12 Apr 2012 20:04:28 GMT
Content-Length: 0
Server: AmazonS3
```

## Sample Request: Access permissions specified using headers

The following request uses ACL-specific request headers to grant the following permissions:

- Write permission to the Amazon S3 LogDelivery group and an AWS account identified by the email xyz@amazon.com.

- Read permission to the Amazon S3 AllUsers group

```
PUT ?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Sun, 29 Apr 2012 22:00:57 GMT
x-amz-grant-write: uri="http://acs.amazonaws.com/groups/s3/LogDelivery",
    emailAddress="xyz@amazon.com"
x-amz-grant-read: uri="http://acs.amazonaws.com/groups/global/AllUsers"
Accept: */
Authorization: authorization string
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: 0w9iImt23VF9s6QofOTDzelF7mrryz7d04Mw23FQCi4O205Zw28Zn+d340/RytoQ
x-amz-request-id: A6A8F01A38EC7138
Date: Sun, 29 Apr 2012 22:01:10 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 227\)](#)
- [DELETE Bucket \(p. 78\)](#)
- [GET Object ACL \(p. 361\)](#)

# PUT Bucket analytics

## Description

This implementation of the `PUT` operation adds an analytics configuration (identified by the analytics ID) to the bucket. You can have up to 1,000 analytics configurations per bucket.

You can choose to have storage class analysis export analysis reports to a comma-separated values (CSV) flat file, see the `DataExport` request element. Reports are updated daily and are based on the object filters you configure. When selecting data export you specify a destination bucket and optional destination prefix where the file is written. You can export the data to a destination bucket in a different account. However, the destination bucket must be in the same region as the bucket that you are making the `PUT` analytics configuration to. For more information, see [Amazon S3 Analytics – Storage Class Analysis](#) in the *Amazon Simple Storage Service Developer Guide*.

### Important

You must create a bucket policy on the destination bucket where the exported file is written to grant permissions to Amazon S3 to write objects to the bucket. For an example policy, see [Granting Permissions for Amazon S3 Inventory and Storage Class Analysis](#).

To use this operation, you must have permissions to perform the `s3:PutAnalyticsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?analytics&id=configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Analytics configuration in the request body
```

### Request Parameters

This implementation of `PUT` uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID identifying the analytics configuration. This ID must match the request element <code>id</code> . Limited to 64 characters.  Type: String  Default: None  Valid Characters for <code>id</code> : a-z A-Z 0-9 - _ .	Yes

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

In the request, you must specify the analytics configuration in the request body, which is specified as XML. The Examples section shows an example of an analytics configuration.

The following table describes the XML elements in the analytics configuration:

Name	Description	Required
<code>AnalyticsConfiguration</code>	<p>Contains the configuration and any analyses for the analytics filter.</p> <p>Type: Container</p> <p>Children: <code>Id</code>, <code>Filter</code>, <code>StorageClassAnalysis</code></p> <p>Ancestor: None</p>	Yes
<code>And</code>	<p>A conjunction (logical AND) of predicates, which is used in evaluating an analytics filter. The operator must have at least two predicates.</p> <p>Type: String</p> <p>Children: <code>Prefix</code>, <code>Tag</code></p> <p>Ancestor: <code>Filter</code></p>	No
<code>Bucket</code>	<p>The Amazon Resource Name (ARN) of the bucket where analytics results are published. This destination bucket must be in the same region as the bucket used for the analytics configuration PUT.</p> <p>Type: String</p> <p>Ancestor: <code>S3BucketDestination</code></p>	Yes
<code>BucketAccountId</code>	<p>The ID of the account that owns the destination bucket where the analytics is published.</p> <p>Although optional, we recommend that the value be set to prevent problems if the destination bucket ownership changes.</p> <p>Type: String</p> <p>Ancestor: <code>S3BucketDestination</code></p>	No
<code>DataExport</code>	<p>A container used to describe how data related to the storage class analysis should be exported.</p> <p>Type: Container</p> <p>Children: <code>OutputSchemaVersion</code>, <code>Destination</code></p>	No

Name	Description	Required
	Ancestor: StorageClassAnalysis	
Destination	<p>Contains information about where to publish the analytics results.</p> <p>Type: Container</p> <p>Children: S3BucketDestination</p> <p>Ancestor: DataExport</p>	Yes
Filter	<p>Specifies an analytics filter. The analytics only includes objects that meet the filter's criteria. If no filter is specified, all of the contents of the bucket are included in the analysis.</p> <p>Type: Container</p> <p>Children: And</p> <p>Ancestor: AnalyticsConfiguration</p>	No
Format	<p>Specifies the output format of the analytics results. Currently, Amazon S3 supports the comma-separated value (CSV) format.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p> <p>Valid values: CSV</p>	Yes
Id	<p>The ID that identifies the analytics configuration. This ID must match the request parameter <code>id</code>.</p> <p>Type: String</p> <p>Ancestor: AnalyticsConfiguration</p>	Yes
Key	<p>The key for a tag.</p> <p>Type: String</p> <p>Ancestor: Tag</p>	Yes
OutputSchemaVersion	<p>The version of the output schema to use when exporting data. Must be V_1.</p> <p>Type: String</p> <p>Ancestor: DataExport</p> <p>Valid values: v_1</p>	Yes

Name	Description	Required
Prefix	The prefix that an object must have to be included in the analytics results.  Type: String  Ancestor: And	No
Prefix	The prefix that is prepended to all analytics results.  Type: String  Ancestor: S3BucketDestination	No
StorageClassAnalysis	Indicates that data related to access patterns will be collected and made available to analyze the tradeoffs between different storage classes.  Type: Container  Children: DataExport  Ancestor: AnalyticsConfiguration	Yes
S3BucketDestination	Contains the bucket ARN, file format, bucket owner (optional), and prefix (optional) where analytics results are published.  Type: Container  Children: Format, BucketAccountId, Bucket, Prefix  Ancestor: Destination.	Yes
Tag	The tag to use when evaluating an analytics filter.  Type: Container  Children: Key, Value  Ancestor: And	No
Value	The value for a tag.  Type: String  Ancestor: Tag	Yes

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

This implementation of the operation does not return response elements.

## Special Errors

Amazon S3 checks the validity of the proposed `AnalyticsConfiguration` element and verifies whether the proposed configuration is valid when you call the `PUT` operation. The following table lists the errors and possible causes.

HTTP Error	Code	Cause
HTTP 400 Bad Request	InvalidArgument	Invalid argument.
HTTP 400 Bad Request	TooManyConfigurations	You are attempting to create a new configuration but have already reached the 1,000-configuration limit.
HTTP 403 Forbidden	AccessDenied	You are not the owner of the specified bucket, or you do not have the <code>s3:PutAnalyticsConfiguration</code> bucket permission to set the configuration on the bucket.

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Creating an Analytics Configuration

The following PUT request for the bucket `examplebucket` creates a new or replaces an existing analytics configuration with the ID `report1`. The configuration is defined in the request body.

```
PUT /?analytics&id=report1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<AnalyticsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>report1</Id>
  <Filter>
    <And>
      <Prefix>images/</Prefix>
      <Tag>
        <Key>dog</Key>
        <Value>corgi</Value>
      </Tag>
    </And>
  </Filter>
  <StorageClassAnalysis>
    <DataExport>
      <OutputSchemaVersion>V_1</OutputSchemaVersion>
      <Destination>
        <S3BucketDestination>
          <Format>CSV</Format>
          <BucketAccountId>123456789012</BucketAccountId>
          <Bucket>arn:aws:s3:::destination-bucket</Bucket>
        </S3BucketDestination>
      </Destination>
    </DataExport>
  </StorageClassAnalysis>
</AnalyticsConfiguration>
```

```
<Prefix>destination-prefix</Prefix>
</S3BucketDestination>
</Destination>
</DataExport>
</StorageClassAnalysis>
</AnalyticsConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 31 Oct 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [GET Bucket analytics \(p. 126\)](#)
- [DELETE Bucket analytics \(p. 80\)](#)
- [List Bucket Analytics Configurations \(p. 206\)](#)

# PUT Bucket cors

## Description

Sets the `cors` configuration for your bucket. If the configuration exists, Amazon S3 replaces it.

To use this operation, you must be allowed to perform the `s3:PutBucketCORS` action. By default, the bucket owner has this permission and can grant it to others.

You set this configuration on a bucket so that the bucket can service cross-origin requests. For example, you might want to enable a request whose origin is `http://www.example.com` to access your Amazon S3 bucket at `my.example.bucket.com` by using the browser's `XMLHttpRequest` capability.

To enable cross-origin resource sharing (CORS) on a bucket, you add the `cors` subresource to the bucket. The `cors` subresource is an XML document in which you configure rules that identify origins and the HTTP methods that can be executed on your bucket. The document is limited to 64 KB in size. For example, the following `cors` configuration on a bucket has two rules:

- The first `CORSRule` allows cross-origin `PUT`, `POST` and `DELETE` requests whose origin is `https://www.example.com` origins. The rule also allows all headers in a pre-flight `OPTIONS` request through the `Access-Control-Request-Headers` header. Therefore, in response to any pre-flight `OPTIONS` request, Amazon S3 will return any requested headers.
- The second rule allows cross-origin `GET` requests from all the origins. The '\*' wildcard character refers to all origins.

```
<corsConfiguration>
<corsRule>
    <allowedOrigin>http://www.example.com</allowedOrigin>

    <allowedMethod>PUT</allowedMethod>
    <allowedMethod>POST</allowedMethod>
    <allowedMethod>DELETE</allowedMethod>

    <allowedHeader>*</allowedHeader>
</corsRule>
<corsRule>
    <allowedOrigin>*</allowedOrigin>
    <allowedMethod>GET</allowedMethod>
</corsRule>
</corsConfiguration>
```

The `cors` configuration also allows additional optional configuration parameters as shown in the following `cors` configuration on a bucket. For example, this `cors` configuration allows cross-origin `PUT` and `POST` requests from `http://www.example.com`.

```
<corsConfiguration>
<corsRule>
    <allowedOrigin>http://www.example.com</allowedOrigin>
    <allowedMethod>PUT</allowedMethod>
    <allowedMethod>POST</allowedMethod>
    <allowedMethod>DELETE</allowedMethod>
    <allowedHeader>*</allowedHeader>
    <maxAgeSeconds>3000</maxAgeSeconds>
    <exposeHeader>x-amz-server-side-encryption</exposeHeader>
</corsRule>
</corsConfiguration>
```

In the preceding configuration, `CORSRule` includes the following additional optional parameters:

- `MaxAgeSeconds`—Specifies the time in seconds that the browser will cache an Amazon S3 response to a pre-flight OPTIONS request for the specified resource. In this example, this parameter is 3000 seconds. Caching enables the browsers to avoid sending pre-flight OPTIONS request to Amazon S3 for repeated requests.
- `ExposeHeader`—Identifies the response header (in this case `x-amz-server-side-encryption`) that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).

When Amazon S3 receives a cross-origin request (or a pre-flight OPTIONS request) against a bucket, it evaluates the `cors` configuration on the bucket and uses the first `CORSRule` rule that matches the incoming browser request to enable a cross-origin request. For a rule to match, the following conditions must be met:

- The request's `Origin` header must match `AllowedOrigin` elements.
- The request method (for example, GET, PUT, HEAD and so on) or the `Access-Control-Request-Method` header in case of a pre-flight OPTIONS request must be one of the `AllowedMethod` elements.
- Every header specified in the `Access-Control-Request-Headers` request header of a pre-flight request must match an `AllowedHeader` element.

For more information about CORS, go to [Enabling Cross-Origin Resource Sharing](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?cors HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Content-MD5: MD5

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>Origin you want to allow cross-domain requests from</AllowedOrigin>
    <AllowedOrigin>...</AllowedOrigin>
    ...
    <AllowedMethod>HTTP method</AllowedMethod>
    <AllowedMethod>...</AllowedMethod>
    ...
    <MaxAgeSeconds>Time in seconds your browser to cache the pre-flight OPTIONS response for a resource</MaxAgeSeconds>
    <AllowedHeader>Headers that you want the browser to be allowed to send</AllowedHeader>
    <AllowedHeader>...</AllowedHeader>
    ...
    <ExposeHeader>Headers in the response that you want accessible from client application</ExposeHeader>
    <ExposeHeader>...</ExposeHeader>
    ...
  </CORSRule>
  <CORSRule>
    ...
  </CORSRule>
  ...

```

```
</CORSConfiguration>
```

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, go to <a href="#">RFC 1864</a> .  Type: String  Default: None	Yes

## Request Elements

Name	Description	Required
CORSConfiguration	Container for up to 100 CORSRules elements.  Type: Container  Children: CORSRules  Ancestor: None	Yes
CORSRule	A set of origins and methods (cross-origin access that you want to allow). You can add up to 100 rules to the configuration.  Type: Container  Children: AllowedOrigin, AllowedMethod, MaxAgeSeconds, ExposeHeader, ID.  Ancestor: CORSConfiguration	Yes
ID	A unique identifier for the rule. The ID value can be up to 255 characters long. The IDs help you find a rule in the configuration.  Type: String  Ancestor: CORSRule	No
AllowedMethod	An HTTP method that you want to allow the origin to execute.  Each CORSRule must identify at least one origin and one method.  Type: Enum (GET, PUT, HEAD, POST, DELETE)	Yes

Name	Description	Required
	Ancestor: CORSRule	
AllowedOrigin	<p>An origin that you want to allow cross-domain requests from. This can contain at most one * wild character.</p> <p>Each CORSRule must identify at least one origin and one method.</p> <p>The origin value can include at most one '*' wild character. For example, "http://*.example.com". You can also specify only * as the origin value allowing all origins cross-domain access.</p> <p>Type: String</p> <p>Ancestor: CORSRule</p>	Yes
AllowedHeader	<p>Specifies which headers are allowed in a pre-flight OPTIONS request via the <code>Access-Control-Request-Headers</code> header. Each header name specified in the <code>Access-Control-Request-Headers</code> header must have a corresponding entry in the rule. Amazon S3 will send only the allowed headers in a response that were requested.</p> <p>This can contain at most one * wild character.</p> <p>Type: String</p> <p>Ancestor: CORSRule</p>	No
MaxAgeSeconds	<p>The time in seconds that your browser is to cache the preflight response for the specified resource.</p> <p>A CORSRule can have at most one MaxAgeSeconds element.</p> <p>Type: Integer (seconds)</p> <p>Ancestor: CORSRule</p>	No
ExposeHeader	<p>One or more headers in the response that you want customers to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).</p> <p>You add one ExposeHeader element in the rule for each header.</p> <p>Type: String</p> <p>Ancestor: CORSRule</p>	No

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

The following examples add the `cors` subresource to a bucket.

### Example : Configure cors

#### Sample Request

The following PUT request adds the `cors` subresource to a bucket (`examplebucket`).

```
PUT /?cors HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Tue, 21 Aug 2012 17:54:50 GMT
Content-MD5: 8dYiLewFWZyGgV2Q5FNI4W==
Authorization: authorization string
Content-Length: 216

<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>http://www.example.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSec>
    <ExposeHeader>x-amz-server-side-encryption</ExposeHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>*</AllowedHeader>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
  </CORSRule>
</CORSConfiguration>
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: CCshOvbOPfxzhwOAdyC4qHj/Ck3F9Q0viXKw3rivZ+GcBoZSOOahvEJfPisZB7B
x-amz-request-id: BDC4B83DF5096BBE
Date: Tue, 21 Aug 2012 17:54:50 GMT
Server: AmazonS3
```

## Related Resources

- [GET Bucket cors \(p. 131\)](#)
- [DELETE Bucket cors \(p. 82\)](#)

- [OPTIONS object \(p. 382\)](#)

# PUT Bucket encryption

## Description

This implementation of the `PUT` operation uses the `encryption` subresource to set the default encryption state of an existing bucket.

This implementation of the `PUT` operation sets default encryption for a buckets using server-side encryption with Amazon S3-managed keys SSE-S3 or AWS KMS-managed Keys (SSE-KMS) bucket. For information about the Amazon S3 default encryption feature, see [Amazon S3 Default Bucket Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

### Important

This operation requires AWS Signature Version 4. For more information, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

To use this operation, you must have permissions to perform the `s3:PutEncryptionConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?encryption HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
default encryption configuration in the request body
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Body

In the request, you specify the encryption configuration in the request body. The encryption configuration is specified as XML, as shown in the following examples that show setting encryption using SSE-S3 or SSE-KMS.

The following is an example of the request body for setting SSE-S3.

```
<ServerSideEncryptionConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Rule>
  <ApplyServerSideEncryptionByDefault>
    <sseAlgorithm>AES256</sseAlgorithm>
```

```

        </ApplyServerSideEncryptionByDefault>
    </Rule>
</ServerSideEncryptionConfiguration>
```

The following is an example of the request body for setting SSE-KMS.

```

<ServerSideEncryptionConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Rule>
        <ApplyServerSideEncryptionByDefault>
            < SSEAlgorithm>aws:kms</SSEAlgorithm>
            < KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
        </ApplyServerSideEncryptionByDefault>
    </Rule>
</ServerSideEncryptionConfiguration>
```

The following table describes the XML elements in the encryption configuration:

Name	Description	Required
ApplyServerSideEncryptionByDefault	<p>Container for setting server-side encryption by default.</p> <p>Type: Container</p> <p>Children: SSEAlgorithm, KMSMasterKeyID</p> <p>Ancestor: Rule</p>	No
KMSMasterKeyID	<p>The AWS KMS master key ID used for the SSE-KMS encryption.</p> <p>Type: String</p> <p>Ancestor: ApplyServerSideEncryptionByDefault</p> <p>Constraint: Can only be used when you set the value of SSEAlgorithm as aws:kms. The default aws/s3 AWS KMS master key is used if this element is absent while the SSEAlgorithm is aws:kms.</p>	No
Rule	<p>Container for server-side encryption by default configuration.</p> <p>Type: Container</p> <p>Children: ApplyServerSideEncryptionByDefault</p> <p>Ancestor: ServerSideEncryptionConfiguration</p>	Yes
ServerSideEncryptionConfiguration	<p>Container for the server-side encryption by default configuration rule.</p> <p>Type: Container</p> <p>Children: Rule</p> <p>Ancestor: None</p>	Yes

Name	Description	Required
SSEAlgorithm	<p>The server-side encryption algorithm to use.</p> <p>Type: String</p> <p>Valid Values: AES256, aws:kms</p> <p>Ancestor: <code>ApplyServerSideEncryptionByDefault</code></p> <p>Constraint: Can only be used when you use <code>ApplyServerSideEncryptionByDefault</code>.</p>	Yes

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Set the Default Encryption Configuration for an S3 Bucket

The following is an example of a `PUT /?encryption` request that specifies to use AWS KMS encryption.

```

PUT /?encryption HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Wed, 06 Sep 2017 12:00:00 GMT
Authorization: authorization string
Content-Length: length

<ServerSideEncryptionConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Rule>
    <ApplyServerSideEncryptionByDefault>
      < SSEAlgorithm>aws:kms</SSEAlgorithm>
      < KMSMasterKeyID>arn:aws:kms:us-east-1:1234/5678example</KMSMasterKeyID>
    </ApplyServerSideEncryptionByDefault>
  </Rule>
</ServerSideEncryptionConfiguration>

```

The following is an example response:

```
HTTP/1.1 100 Continue
```

```
HTTP/1.1 200 OK
x-amz-id-2: B3Z1w/R0GaUCDHStDVuoz+4NSndjUDYuE3jvJ5kvrDroucdFCygEQYEwpC0Lj0Cv
x-amz-request-id: E0DE682C2FDDBCF8
Date: Wed, 06 Sep 2017 12:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [GET Bucket encryption \(p. 135\)](#)
- [DELETE Bucket encryption \(p. 84\)](#)

# PUT Bucket inventory

## Description

This implementation of the `PUT` operation adds an inventory configuration (identified by the inventory ID) to the bucket. You can have up to 1,000 inventory configurations per bucket.

Amazon S3 inventory generates inventories of the objects in the bucket on a daily or weekly basis, and the results are published to a flat file. The bucket that is inventoried is called the *source bucket*, and the bucket where the inventory flat file is stored is called the *destination bucket*. The destination bucket must be in the same AWS Region as the source bucket.

When you configure an inventory for a source bucket, you specify the destination bucket where you want the inventory to be stored, and whether to generate the inventory daily or weekly. You can also configure what object metadata to include and whether to inventory all object versions or only current versions. For more information, see [Amazon S3 Inventory](#) in the *Amazon Simple Storage Service Developer Guide*.

### Important

You must create a bucket policy on the destination bucket to grant permissions to Amazon S3 to write objects to the bucket in the defined location. For an example policy, see [Granting Permissions for Amazon S3 Inventory and Storage Class Analysis](#).

To use this operation, you must have permissions to perform the `s3:PutInventoryConfiguration` action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?inventory&id=configuration-ID HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Inventory configuration in the request body
```

### Request Parameters

This implementation of `PUT` uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID identifying the inventory configuration. This ID must match the request element <code>id</code> . Limited to 64 characters.  Type: String  Default: None  Valid Characters for <code>id</code> : a-z A-Z 0-9 - _ .	Yes

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

In the request, you must specify the inventory configuration in the request body, which is specified as XML. The Examples section shows an example of an inventory configuration.

The following table describes the XML elements in the inventory configuration:

Name	Description	Required
AccountId	<p>The ID of the account that owns the destination bucket.</p> <p>Although optional, we recommend that the value be set to prevent problems if the destination bucket ownership changes.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p>	No
Bucket	<p>The Amazon Resource Name (ARN) of the bucket where inventory results are published. This destination bucket must be in the same AWS Region as the source bucket.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p>	Yes
Destination	<p>Contains information about where to publish the inventory results.</p> <p>Type: Container</p> <p>Children: S3BucketDestination</p> <p>Ancestor: InventoryConfiguration</p>	Yes
Encryption	<p>Contains the type of server-side encryption to use to encrypt the inventory.</p> <p>Type: Container</p> <p>Children: SSE-KMS, SSE-S3</p> <p>Ancestor: S3BucketDestination</p>	No
Field	<p>Contains the optional fields that are included in the inventory results. Multiple <code>Field</code> elements can be contained in <code>OptionalFields</code>.</p> <p>Type: String</p> <p>Ancestor: OptionalFields</p>	No

Name	Description	Required
	Valid values: Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode, ObjectLockLegalHoldStatus	
Filter	<p>Specifies an inventory filter. The inventory only includes objects that meet the filter's criteria. If no filter is specified, all of the contents of the bucket are inventoried.</p> <p>Type: Container</p> <p>Children: Prefix</p> <p>Ancestor: InventoryConfiguration</p>	No
Format	<p>Specifies the output format of the inventory results. Currently, Amazon S3 supports the comma-separated values (CSV) format, the <a href="#">Apache optimized row columnar (ORC)</a> format, and the <a href="#">Apache Parquet (Parquet)</a> format.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p> <p>Valid values: CSV, ORC, or Parquet</p>	Yes
Frequency	<p>Specifies how frequently inventory results are produced.</p> <p>Type: String</p> <p>Ancestor: Schedule</p> <p>Valid values: Daily, or Weekly</p>	Yes
Id	<p>The ID identifying the inventory configuration. This ID must match the request parameter id.</p> <p>Type: String</p> <p>Ancestor: InventoryConfiguration</p>	Yes
IncludedObjectVersions	<p>Specifies which object versions to include in the inventory results. If set to All, the list includes all of the object versions, which adds the version-related fields VersionId, IsLatest, and DeleteMarker to the list. If set to Current, the list does not contain these version-related fields.</p> <p>Type: String</p> <p>Ancestor: InventoryConfiguration</p> <p>Valid values: Current or All</p>	Yes

Name	Description	Required
InventoryConfiguration	<p>Contains the inventory configuration.</p> <p>Type: Container</p> <p>Children: Id, IsEnabled, Filter, Destination, Schedule, IncludedObjectVersions, and OptionalFields elements.</p> <p>Ancestor: None</p>	Yes
IsEnabled	<p>Specifies whether the inventory is enabled or disabled.</p> <p>If set to <code>True</code>, inventory results are generated. If set to <code>False</code>, no inventory is generated.</p> <p>Type: String</p> <p>Ancestor: InventoryConfiguration</p> <p>Valid values: <code>True</code> or <code>False</code></p>	Yes
KeyId	<p>The AWS KMS customer master key (CMK) used to encrypt the inventory file.</p> <p>Type: String</p> <p>Ancestor: SSE-KMS</p> <p>Valid values: ARN of the CMK</p>	No
OptionalFields	<p>Contains the optional fields that are included in the inventory results.</p> <p>Type: Container</p> <p>Children: Field</p> <p>Ancestor: InventoryConfiguration</p>	No
Prefix	<p>The prefix that an object must have to be included in the inventory results.</p> <p>Type: String</p> <p>Ancestor: Filter</p>	No
Prefix	<p>The prefix that is prepended to all inventory results.</p> <p>Type: String</p> <p>Ancestor: S3BucketDestination</p>	No

Name	Description	Required
Schedule	<p>Contains the frequency for generating inventory results.</p> <p>Type: Container</p> <p>Children: Frequency</p> <p>Ancestor: Destination</p>	Yes
SSE-KMS	<p>Specifies to use server-side encryption with AWS KMS-managed keys (SSE-KMS) and contains the key that is used to encrypt the inventory file.</p> <p>Type: Container</p> <p>Children: KeyId</p> <p>Ancestor: Encryption</p>	No
SSE-S3	<p>Specifies to use server-side encryption with Amazon S3-managed keys (SSE-S3) to encrypt the inventory file.</p> <p>Type: Container</p> <p>Ancestor: Encryption</p> <p>Valid values: empty</p>	No
S3BucketDestination	<p>Contains the bucket ARN, file format, bucket owner (optional), prefix where inventory results are published (optional), and the type of server-side encryption that is used to encrypt the file (optional).</p> <p>Type: Container</p> <p>Children: Format, AccountId, Bucket, Prefix, Encryption</p> <p>Ancestor: Destination</p>	Yes

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Special Errors

Amazon S3 checks the validity of the proposed `InventoryConfiguration` element and verifies whether the proposed configuration is valid when you call the `PUT` operation. The following table lists the errors and possible causes.

HTTP Error	Code	Cause
HTTP 400 Bad Request	InvalidArgument	Invalid argument.
HTTP 400 Bad Request	TooManyConfigurations	You are attempting to create a new configuration but have already reached the 1,000-configuration limit.
HTTP 403 Forbidden	AccessDenied	You are not the owner of the specified bucket, or you do not have the <code>s3:PutInventoryConfiguration</code> bucket permission to set the configuration on the bucket.

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Creating an Inventory Configuration

The following `PUT` request for the bucket `examplebucket` creates a new or replaces an existing inventory configuration with the ID `report1`. The configuration is defined in the request body.

```

PUT /?inventory&id=report1 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Id>report1</Id>
    <IsEnabled>true</IsEnabled>
    <Filter>
        <Prefix>filterPrefix/</Prefix>
    </Filter>
    <Destination>
        <S3BucketDestination>
            <Format>CSV</Format>
            <AccountId>123456789012</AccountId>
            <Bucket>arn:aws:s3:::destination-bucket</Bucket>
            <Prefix>prefix1</Prefix>
            <Encryption>
                <SSE-KMS>
                    <KeyId>arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab</KeyId>
                </SSE-KMS>
            </Encryption>
        </S3BucketDestination>
    </Destination>
    <Schedule>
        <Frequency>Daily</Frequency>
    </Schedule>
    <IncludedObjectVersions>All</IncludedObjectVersions>
</InventoryConfiguration>
```

```
<OptionalFields>
  <Field>Size</Field>
  <Field>LastModifiedDate</Field>
  <Field>ETag</Field>
  <Field>StorageClass</Field>
  <Field>IsMultipartUploaded</Field>
  <Field>ReplicationStatus</Field>
  <Field>EncryptionStatus</Field>
  <Field>ObjectLockRetainUntilDate</Field>
  <Field>ObjectLockMode</Field>
  <Field>ObjectLockLegalHoldStatus</Field>
</OptionalFields>
</InventoryConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Mon, 31 Oct 2016 12:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [GET Bucket Inventory \(p. 139\)](#)
- [DELETE Bucket inventory \(p. 86\)](#)
- [List Bucket Inventory Configurations \(p. 210\)](#)

# PUT Bucket lifecycle

## Description

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, go to [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

### Note

Bucket lifecycle configuration now supports specifying a lifecycle rule using an object key name prefix, one or more object tags, or a combination of both. Accordingly, this section describes the latest API. The previous version of the API supported filtering based only on an object key name prefix, which is supported for backward compatibility. For the related API description, see [PUT Bucket lifecycle \(Deprecated\) \(p. 569\)](#).

## Permissions

By default, all Amazon S3 resources are private, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration). Only the resource owner (that is, the AWS account that created it) can access the resource. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, a user must get the `s3:PutLifecycleConfiguration` permission.

You can also explicitly deny permissions. Explicit deny also supersedes any other permissions. If you want to block users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

- `s3:DeleteObject`
- `s3:DeleteObjectVersion`
- `s3:PutLifecycleConfiguration`

For more information about permissions, see [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5

Lifecycle configuration in the request body
```

For details about *authorization string*, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

### Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

Name	Description	Required
Content-MD5	<p>The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, see <a href="#">RFC 1864</a>.</p> <p>Type: String</p> <p>Default: None</p>	Yes

## Request Body

You specify the lifecycle configuration in your request body. The lifecycle configuration is specified as XML consisting of one or more rules.

```
<LifecycleConfiguration>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
  </Rule>
  ...
</LifecycleConfiguration>
```

Each rule consists of the following:

- Filter identifying a subset of objects to which the rule applies. The filter can be based on a key name prefix, object tags, or a combination of both.
- Status whether the rule is in effect.
- One or more lifecycle transition and expiration actions that you want Amazon S3 to perform on the objects identified by the filter. If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object (one current version and zero or more noncurrent versions). Amazon S3 provides predefined actions that you can specify for current and noncurrent object versions.

For example,

```
<LifecycleConfiguration>
  <Rule>
    <Filter>
      <Prefix>key-prefix</Prefix>
    </Filter>
    <Status>rule-status</Status>
    One or more Transition/Expiration lifecycle actions.
  </Rule>
</LifecycleConfiguration>
```

For more information, see [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information, see [Lifecycle Configuration Elements](#) in the *Amazon Simple Storage Service Developer Guide*.

The following table describes the XML elements in the lifecycle configuration:

Name	Description	Required
AbortIncompleteMultipartUpload	<p>Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.</p> <p>When you specify this lifecycle action, the rule cannot specify a tag-based filter.</p> <p>For more information, see <a href="#">Lifecycle Configuration Elements</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Child: <code>DaysAfterInitiation</code></p> <p>Type: Container</p> <p>Ancestor: Rule.</p>	Yes, if no other action is specified for the rule.
And	<p>Container for specifying rule filters. These filters determine the subset of objects to which the rule applies.</p> <p>Type: String</p> <p>Ancestor: Rule</p>	Yes, if you specify more than one filter condition (for example, one prefix and one or more tags).
Date	<p>Date when you want Amazon S3 to take the action. For more information, see <a href="#">Lifecycle Rules: Based on a Specific Date</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The date value must conform to the ISO 8601 format. The time is always midnight UTC.</p> <p>Type: String</p> <p>Ancestor: Expiration or Transition</p>	Yes, if Days and ExpiredObjectDeleteMarker are absent.
Days	<p>Specifies the number of days after object creation when the specific rule action takes effect.</p> <p>Type: Nonnegative Integer when used with Transition, Positive Integer when used with Expiration.</p> <p>Ancestor: Expiration, Transition.</p>	Yes, if Date and ExpiredObjectDeleteMarker are absent.
DaysAfterInitiation	Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible	Yes, if ancestor is specified.

Name	Description	Required
	<p>for an abort operation and Amazon S3 aborts the incomplete multipart upload.</p> <p>Type: Positive Integer.</p> <p>Ancestor: <code>AbortIncompleteMultipartUpload</code>.</p>	
<code>Expiration</code>	<p>This action specifies a period in an object's lifetime when Amazon S3 should take the appropriate expiration action. The action Amazon S3 takes depends on whether the bucket is versioning-enabled.</p> <ul style="list-style-type: none"> <li>• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</li> <li>• Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. A versioning-enabled bucket can have many versions of the same object, one current version, and zero or more noncurrent versions.</li> </ul> <p>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</p> <p><b>Important</b> If your bucket state is versioning-suspended, Amazon S3 creates a delete marker with version ID <code>null</code>. If you have a version with version ID <code>null</code>, then Amazon S3 overwrites that version.</p> <p><b>Note</b> To set expiration for noncurrent objects, you must use the <code>NoncurrentVersionExpiration</code> action.</p> <p>Type: Container</p> <p>Children: Days or Date</p> <p>Ancestor: <code>Rule</code></p>	Yes, if no other action is present in the <code>Rule</code> .

Name	Description	Required
Filter	<p>Container for elements that describe the filter identifying a subset of objects to which the lifecycle rule applies. If you specify an empty filter (&lt;Filter&gt;/&lt;/Filter&gt;), the rule applies to all objects in the bucket.</p> <p>Type: String</p> <p>Children: Prefix, Tag</p> <p>Ancestor: Rule</p>	Yes
ID	<p>Unique identifier for the rule. The value cannot be longer than 255 characters.</p> <p>Type: String</p> <p>Ancestor: Rule</p>	No
Key	<p>Specifies the key of a tag. A tag key can be up to 128 Unicode characters in length.</p> <p>Tag keys that you specify in a lifecycle rule filter must be unique.</p> <p>For more information, see <a href="#">Object Tagging</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Ancestor: Tag</p>	Yes, if <Tag> parent is specified.
LifecycleConfiguration	<p>Container for lifecycle rules. You can add as many as 1,000 rules.</p> <p>Type: Container</p> <p>Children: Rule</p> <p>Ancestor: None</p>	Yes

Name	Description	Required
ExpiredObjectDeleteMarker	<p>On a versioned bucket (versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to direct Amazon S3 to delete expired object delete markers. For an example, see <a href="#">Example 7: Removing Expired Object Delete Markers</a> in the <i>Amazon Simple Storage Service Developer Guide</i>. On a nonversioned bucket, adding this element in a policy is meaningless because you cannot have delete markers and the element doesn't do anything.</p> <p>For more information, see <a href="#">Lifecycle Configuration Elements</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>When you specify this lifecycle action, the rule cannot specify a tag-based filter.</p> <p>Type: String</p> <p>Valid values: <code>true</code>   <code>false</code> (the value <code>false</code> is allowed, but it is no-op and Amazon S3 does not take action if the value is <code>false</code>)</p> <p>Ancestor: <code>Expiration</code>.</p>	Yes, if Date and Days are absent.
NoncurrentDays	<p>Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see <a href="#">How Amazon S3 Calculates When an Object Became Noncurrent</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Nonnegative Integer when used with <code>NoncurrentVersionTransition</code>, Positive Integer when used with <code>NoncurrentVersionExpiration</code>.</p> <p>Ancestor: <code>NoncurrentVersionExpiration</code> or <code>NoncurrentVersionTransition</code></p>	Yes

Name	Description	Required
NoncurrentVersionExpiration	<p>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.</p> <p>You set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: <code>NoncurrentDays</code></p> <p>Ancestor: <code>Rule</code></p>	Yes, if no other action is present in the <code>Rule</code> .
NoncurrentVersionTransition	<p>Container for the transition rule that describes when noncurrent objects transition to the <code>STANDARD_IA</code>, <code>ONEZONE_IA</code>, or <code>GLACIER</code> storage class.</p> <p>If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request that Amazon S3 transition noncurrent object versions at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: <code>NoncurrentDays</code> and <code>StorageClass</code></p> <p>Ancestor: <code>Rule</code></p>	Yes, if no other action is present in the <code>Rule</code> .
Prefix	<p>Object key prefix identifying one or more objects to which the rule applies. Empty prefix (<code>&lt;Prefix&gt;&lt;/Prefix&gt;</code>) indicates there is no filter based on key prefix.</p> <p>There can be at most one <code>Prefix</code> in a lifecycle rule <code>Filter</code>.</p> <p>Type: String</p> <p>Ancestor: <code>Filter</code> or <code>And</code> (if you specify multiple filters such as a prefix and one or more tags).</p>	No
Rule	<p>Container for a lifecycle rule. A lifecycle configuration can contain as many as 1,000 rules.</p> <p>Type: Container</p> <p>Ancestor: <code>LifecycleConfiguration</code></p>	Yes

Name	Description	Required
Status	<p>If Enabled, Amazon S3 executes the rule as scheduled. If Disabled, Amazon S3 ignores the rule.</p> <p>Type: String</p> <p>Ancestor: Rule</p> <p>Valid values: Enabled, Disabled.</p>	Yes
StorageClass	<p>Specifies the Amazon S3 storage class to which you want the object to transition.</p> <p>Type: String</p> <p>Ancestor: Transition and NoncurrentVersionTransition</p> <p>Valid values: STANDARD_IA   ONEZONE_IA   GLACIER.</p>	Yes  This element is required only if you specify one or both its ancestors.
Tag	<p>Container for specifying a tag key and value. Each tag has a key and a value.</p> <p>Type: Container</p> <p>Children: Key and Value</p> <p>Ancestor: Filter or And (if you specify multiple filters such as a prefix and one or more tags).</p>	No

Name	Description	Required
Transition	<p>This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or the GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.</p> <ul style="list-style-type: none"> <li>If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.</li> <li>Otherwise, when your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.</li> </ul> <p><b>Note</b> A versioning-enabled bucket can have many versions of an object. This action has no impact on the noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.</p> <p>Type: Container Children: Days or Date, and StorageClass Ancestor: Rule</p>	Yes, if no other action is present in the Rule.
Value	<p>Specifies the value for a tag key. Each object tag is a key-value pair.</p> <p>Tag value can be up to 256 Unicode characters in length.</p> <p>Type: String Ancestor: Tag</p>	Yes, if <Tag> parent is specified.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Add lifecycle configuration - bucket not versioning-enabled

The following lifecycle configuration specifies two rules, each with one action.

- The `Transition` action requests Amazon S3 to transition objects with the "documents/" prefix to the `GLACIER` storage class 30 days after creation.
- The `Expiration` action requests Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.

```
<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample `PUT /?lifecycle` request that adds the preceding lifecycle configuration to the `examplebucket` bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:11:21 GMT
Content-MD5: q6yJD1IkBaGGfb3QLY69A==
Authorization: authorization string
Content-Length: 415

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
```

```
</Filter>
<Status>Enabled</Status>
<Transition>
  <Days>30</Days>
  <StorageClass>GLACIER</StorageClass>
</Transition>
</Rule>
<Rule>
  <ID>id2</ID>
  <Filter>
    <Prefix>logs/</Prefix>
  </Filter>
  <Status>Enabled</Status>
  <Expiration>
    <Days>365</Days>
  </Expiration>
</Rule>
</LifecycleConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJ0JJYcd+1j5nM/rUFiiiz/fNbDOsd3JUE8NWMLNHXmvPfwMpdc
x-amz-request-id: 9E26D08072A8EF9E
Date: Wed, 14 May 2014 02:11:22 GMT
Content-Length: 0
Server: AmazonS3
```

## Example 2: Add lifecycle configuration - bucket is versioning-enabled

The following lifecycle configuration specifies two rules, each with one action for Amazon S3 to perform. You specify these actions when your bucket is versioning-enabled or versioning is suspended:

- The `NoncurrentVersionExpiration` action requests Amazon S3 to expire noncurrent versions of objects with the "logs/" prefix 100 days after the objects become noncurrent.
- The `NoncurrentVersionTransition` action requests Amazon S3 to transition noncurrent versions of objects with the "documents/" prefix to the GLACIER storage class 30 days after they become noncurrent.

```
<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>100</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```

```
<StorageClass>GLACIER</StorageClass>
</NoncurrentVersionTransition>
</Rule>
</LifeCycleConfiguration>
```

The following is a sample `PUT /?lifecycle` request that adds the preceding lifecycle configuration to the `examplebucket` bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mDqVNKkaZDddgnw==
Authorization: authorization string
Content-Length: 598

<LifeCycleConfiguration>
  <Rule>
    <ID>DeleteAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>1</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
  <Rule>
    <ID>TransitionSoonAfterBecomingNonCurrent</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>0</NoncurrentDays>
      <StorageClass>GLACIER</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
</LifeCycleConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrmMmoO//3bMdDTw/CnjArwje+J49Hf+j44yRb/VmbIkgl05A+PT98Cp/6k07hf+LD2mY=
x-amz-request-id: 02D7EC4C10381EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3
```

## Additional Examples

Lifecycle configuration topic in the developer guide provides additional examples. For more information, go to [Examples of Lifecycle Configuration](#).

## Related Resources

- [GET Bucket lifecycle \(p. 145\)](#)
- [DELETE Bucket lifecycle \(p. 88\)](#)

# PUT PublicAccessBlock

## Description

This operation creates or modifies the `PublicAccessBlock` configuration for an Amazon S3 bucket. In order to use this operation, you must have the `s3:PutBucketPublicAccessBlock` permission. For more information about Amazon S3 permissions, see [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

### Important

When Amazon S3 evaluates the `PublicAccessBlock` configuration for a bucket or an object, it checks the `PublicAccessBlock` configuration for both the bucket (or the bucket that contains the object) and the bucket owner's account. If the `PublicAccessBlock` configurations are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

For more information about when Amazon S3 considers a bucket or an object public, see [The Meaning of "Public"](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization string> (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This operation uses the following request elements. You can enable `BlockPublicAcls`, `IgnorePublicAcls`, `BlockPublicPolicy`, and `RestrictPublicBuckets` in any combination.

Name	Description	Required
<code>PublicAccessBlock</code>	A key that indicates the <code>Block</code> configuration.  Type: Container  Children: <code>BlockPublicAcls</code> , <code>IgnorePublicAcls</code> , <code>BlockPublicPolicy</code> , <code>RestrictPublicBuckets</code>	Yes
<code>BlockPublicAcls</code>	Specifies whether Amazon S3 should block public access control lists (ACLs) for this bucket. Setting this element to TRUE causes the following behavior:	No

Name	Description	Required
	<ul style="list-style-type: none"> <li>• <a href="#">PUT Bucket acl (p. 235)</a> and <a href="#">PUT Object acl (p. 447)</a> calls fail if the specified ACL is public.</li> <li>• <a href="#">PUT Object (p. 412)</a> calls fail if the request includes a public ACL.</li> </ul> <p><b>Important</b> Enabling this setting doesn't affect existing policies or ACLs.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: <code>TRUE   FALSE</code></p>	
<code>IgnorePublicAcls</code>	<p>Specifies whether Amazon S3 should ignore public ACLs for this bucket. Setting this element to <code>TRUE</code> causes Amazon S3 to ignore all public ACLs on this bucket and any objects that it contains.</p> <p><b>Important</b> Enabling this setting doesn't affect the persistence of any existing ACLs and doesn't prevent new public ACLs from being set.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: <code>TRUE   FALSE</code></p>	No
<code>BlockPublicPolicy</code>	<p>Specifies whether Amazon S3 should block public bucket policies for this bucket. Setting this element to <code>TRUE</code> causes Amazon S3 to reject calls to <a href="#">PUT Bucket policy (p. 300)</a> if the specified policy allows public access.</p> <p><b>Important</b> Enabling this setting doesn't affect existing bucket policies.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: <code>TRUE   FALSE</code></p>	No
<code>RestrictPublicBuckets</code>	<p>Specifies whether Amazon S3 should restrict public bucket policies for this bucket. If this element is set to <code>TRUE</code>, then only AWS services and authorized users within the bucket owner's account can access this bucket if it has a public bucket policy.</p> <p><b>Important</b> Enabling this setting doesn't affect previously stored bucket policies, except that public and cross-account access within any public bucket policy, including non-public delegation to specific accounts, is blocked.</p> <p>Type: Boolean</p> <p>Ancestor: <code>PublicAccessBlockConfiguration</code></p> <p>Valid values: <code>TRUE   FALSE</code></p>	No

# Responses

## Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

This operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

# Examples

## First Sample Request

The following request puts a bucket PublicAccessBlock configuration that rejects public ACLs.

```
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

<?xml version="1.0" encoding="UTF-8"?>
<PublicAccessBlockConfiguration>
    <BlockPublicAcls>TRUE</BlockPublicAcls>
    <IgnorePublicAcls>FALSE</IgnorePublicAcls>
    <BlockPublicPolicy>FALSE</BlockPublicPolicy>
    <RestrictPublicBuckets>FALSE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

## First Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0
```

## Second Sample Request

The following request puts a bucket PublicAccessBlock configuration that ignores public ACLs and restricts access to public buckets.

```
PUT /<bucket-name>?publicAccessBlock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
x-amz-date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <signatureValue>

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

```
<PublicAccessBlockConfiguration>
  <BlockPublicAcls>FALSE</BlockPublicAcls>
  <IgnorePublicAcls>TRUE</IgnorePublicAcls>
  <BlockPublicPolicy>FALSE</BlockPublicPolicy>
  <RestrictPublicBuckets>TRUE</RestrictPublicBuckets>
</PublicAccessBlockConfiguration>
```

## Second Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4REXAMPLEPi4hklTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
Content-Length: 0
```

## Related Resources

- [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service Developer Guide*.
- [GET PublicAccessBlock \(p. 153\)](#)
- [DELETE PublicAccessBlock \(p. 89\)](#)
- [GET BucketPolicyStatus \(p. 170\)](#)
- [GET PublicAccessBlock \(p. 69\)](#)
- [PUT PublicAccessBlock \(p. 72\)](#)
- [DELETE PublicAccessBlock \(p. 68\)](#)

# PUT Bucket logging

## Description

This implementation of the `PUT` operation uses the `logging` subresource to set the logging parameters for a bucket and to specify permissions for who can view and modify the logging parameters. All logs are saved to buckets in the same AWS Region as the source bucket. To set the logging status of a bucket, you must be the bucket owner.

The bucket owner is automatically granted `FULL_CONTROL` to all logs. You use the `Grantee` request element to grant access to other people. The `Permissions` request element specifies the kind of access the grantee has to the logs.

To enable logging, you use `LoggingEnabled` and its children request elements. To disable logging, you use an empty `BucketLoggingStatus` request element:

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01" />
```

For more information about server access logging, see [Server Access Logging](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information about creating a bucket, see [PUT Bucket \(p. 227\)](#). For more information about returning the logging status of a bucket, see [GET Bucket logging \(p. 157\)](#).

## Requests

### Syntax

```
PUT /?logging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

*Request elements vary depending on what you're setting.*

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

Name	Description	Required
<code>BucketLoggingStatus</code>	Container for logging status information.  Type: Container  Children: <code>LoggingEnabled</code>	Yes

Name	Description	Required
	Ancestry: None	
EmailAddress	<p>Email address of the person being granted logging permissions.</p> <p>Type: String</p> <p>Children: None</p> <p>Ancestry: <code>BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant.Grantee</code></p>	No
Grant	<p>Container for the grantee and his/her logging permissions.</p> <p>Type: Container</p> <p>Children: Grantee, Permission</p> <p>Ancestry: <code>BucketLoggingStatus.LoggingEnabled.TargetGrants</code></p>	No
Grantee	<p>Container for <code>EmailAddress</code> of the person being granted logging permissions. For more information, see <a href="#">Grantee Values (p. 283)</a>.</p> <p>Type: Container</p> <p>Children: <code>EmailAddress</code></p> <p>Ancestry: <code>BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant</code></p>	No
LoggingEnabled	<p>Container for logging information. This element is present when you are enabling logging (and not present when you are disabling logging).</p> <p>Type: Container</p> <p>Children: Grant, TargetBucket, TargetPrefix</p> <p>Ancestry: <code>BucketLoggingStatus</code></p>	No
Permission	<p>Logging permissions given to the Grantee for the bucket. The bucket owner is automatically granted FULL_CONTROL to all logs delivered to the bucket. This optional element enables you to grant access to others.</p> <p>Type: String</p> <p>Valid Values: FULL_CONTROL   READ   WRITE</p> <p>Children: None</p> <p>Ancestry: <code>BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant</code></p>	No

Name	Description	Required
TargetBucket	<p>Specifies the bucket where you want Amazon S3 to store server access logs, which is the target bucket. The bucket that's being logged is the source bucket. The target bucket can be any bucket that you own that is in the same Region as the source bucket, including the source bucket itself. You can also configure multiple buckets to deliver their logs to the same target bucket. In this case, you should choose a different TargetPrefix for each source bucket so that the delivered log files can be distinguished by key.</p> <p>Type: String</p> <p>Children: None</p> <p>Ancestry: BucketLoggingStatus.LoggingEnabled</p>	No
TargetGrants	<p>Container for granting information.</p> <p>Type: Container</p> <p>Children: Grant, Permission</p> <p>Ancestry: BucketLoggingStatus.LoggingEnabled</p>	No
TargetPrefix	<p>This element lets you specify a prefix for the keys that the log files will be stored under.</p> <p>Type: String</p> <p>Children: None</p> <p>Ancestry: BucketLoggingStatus.LoggingEnabled</p>	Yes, if the TargetBucket element is specified.

## Grantee Values

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By the person's ID:

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

DisplayName is optional and ignored in the request.

- By Email address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="AmazonCustomerByEmail"><EmailAddress><replaceable>Grantees@email.com</
replaceable></EmailAddress><lt;/Grantee>
```

The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By URI:

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

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request enables logging and gives the grantee of the bucket READ access to the logs.

```
PUT ?logging HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access_log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress>
        </Grantee>
        <Permission>READ</Permission>
      </Grant>
    </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
```

## Sample Request Disabling Logging

This request disables logging on the bucket, quotes.

```
PUT ?logging HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: authorization string

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01" />
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
```

## Related Resources

- [PUT Object \(p. 412\)](#)
- [DELETE Bucket \(p. 78\)](#)
- [PUT Bucket \(p. 227\)](#)
- [GET Bucket logging \(p. 157\)](#)

# PUT Bucket metrics

## Description

Sets or updates a metrics configuration for the CloudWatch request metrics (specified by the metrics configuration ID) from the bucket. You can have up to 1,000 metrics configurations per bucket. If you're updating an existing metrics configuration, note that this is a full replacement of the existing metrics configuration. If you don't include the elements you want to keep, they are erased.

To use this operation, you must have permissions to perform the `s3:PutMetricsConfiguration` action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about CloudWatch request metrics for Amazon S3, see [Monitoring Metrics with Amazon CloudWatch](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?metrics&id=id HTTP/1.1
HOST: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
```

**Authorization:** *authorization string* (see [Authenticating Requests \(AWS Signature Version 4\)](#))  
**Metrics configuration in the request body.**

## Request Parameters

This implementation of PUT uses the parameter in the following table.

Parameter	Description	Required
<code>id</code>	The ID used to identify the metrics configuration.	Yes

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

In the request, you must specify the metrics configuration in the request body, which is specified as XML. The Examples section shows an example of a metrics configuration.

The following table describes the XML elements in the metrics configuration:

Parameter	Description	Required
<code>And</code>	A conjunction (logical AND) of predicates, which is used in evaluating a metrics filter. The operator must have at least two predicates in any combination, and an object must match all of the predicates for the filter to apply.  Type: Container  Children: <code>Prefix</code> , <code>Tag</code>  Ancestor: <code>Filter</code>	No
<code>Filter</code>	Specifies a metrics configuration filter. The metrics configuration includes only objects that meet the filter's criteria. A filter must be a prefix, a tag, or a conjunction (And). There's a limit of 11 predicates for each filter, of which there can be one prefix and up to ten tags in a single filter.  Type: Container  Children: <code>And</code>	No
<code>Id</code>	The ID used to identify the metrics configuration.  Type: String  Ancestor: <code>MetricsConfiguration</code>	Yes
<code>Key</code>	The name of the tag.	No

Parameter	Description	Required
	Type: String  Ancestor: Tag	
MetricsConfiguration	Specifies the metrics configuration for CloudWatch request metrics on this bucket.  Type: Container  Ancestor: None	Yes
Prefix	The prefix that an object must have to be included in the metrics results.  Type: String  Ancestor: And	No
Tag	A key-value name pair, used to organize objects by association.  Type: Container  Children: Key, Value,  Ancestor: And	No
Value	The value of the tag.  Type: String  Ancestor: Tag	No

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

Amazon S3 checks the validity of the proposed MetricsConfiguration element and verifies whether the proposed configuration is valid when you call the PUT operation. The following table lists the errors and possible causes.

HTTP Error	Code	Cause
HTTP 400 Bad Request	TooManyConfigurations	You are attempting to create a new configuration but have already reached the 1,000-configuration limit.

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### First Sample Request

Put a metric configuration that enables metrics for an entire bucket.

```
PUT /?metrics&id=EntireBucket HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:21 GMT
Authorization: signatureValue
Content-Length: 159

<?xml version="1.0" encoding="UTF-8"?>
<MetricsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>EntireBucket</Id>
</MetricsConfiguration>
```

### First Sample Response

Put a metric configuration that enables metrics for an entire bucket.

```
HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hk1TXouTf0hccUjo0icPEXAMPLEutBj3M7fPGLWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:22 GMT
Server: AmazonS3
```

### Second Sample Request

Put a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
PUT /?metrics&id=ImportantBlueDocuments HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Thu, 15 Nov 2016 00:17:29 GMT
Authorization: signatureValue
Content-Length: 480

<?xml version="1.0" encoding="UTF-8"?>
<MetricsConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Id>ImportantBlueDocuments</Id>
  <Filter>
    <And>
      <Prefix>documents/</Prefix>
      <Tag>
        <Key>priority</Key>
        <Value>high</Value>
      </Tag>
      <Tag>
        <Key>class</Key>
        <Value>blue</Value>
      </Tag>
    </And>
  </Filter>
</MetricsConfiguration>
```

## Second Sample Response

Put a metrics configuration that enables metrics for objects that start with a particular prefix and also have specific tags applied.

```
HTTP/1.1 204 No Content
x-amz-id-2: ITnGT1y4REXAMPLEPi4hkLTXouTf0hccUjo0iCPEXAMPLEutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991EXAMPLE5321
Date: Thu, 15 Nov 2016 00:17:29 GMT
Server: AmazonS3
```

## Related Resources

- [DELETE Bucket metrics \(p. 90\)](#)
- [GET Bucket metrics \(p. 160\)](#)
- [List Bucket Metrics Configurations \(p. 215\)](#)
- [Monitoring Metrics with Amazon CloudWatch](#) in the *Amazon Simple Storage Service Developer Guide*.

# PUT Bucket notification

## Description

The Amazon S3 notification feature enables you to receive notifications when certain events happen in your bucket. For more information about event notifications, go to [Configuring Event Notifications](#) in the *Amazon Simple Storage Service Developer Guide*.

Using this API, you can replace an existing notification configuration. The configuration is an XML file that defines the event types that you want Amazon S3 to publish and the destination where you want Amazon S3 to publish an event notification when it detects an event of the specified type.

By default, your bucket has no event notifications configured. That is, the notification configuration will be an empty `NotificationConfiguration`.

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

This operation replaces the existing notification configuration with the configuration you include in the request body.

After Amazon S3 receives this request, it first verifies that any Amazon Simple Notification Service (Amazon SNS) or Amazon Simple Queue Service (Amazon SQS) destination exists, and that the bucket owner has permission to publish to it by sending a test notification. In the case of AWS Lambda destinations, Amazon S3 verifies that the Lambda function permissions grant Amazon S3 permission to invoke the function from the Amazon S3 bucket. For more information, go to [Configuring Notifications for Amazon S3 Events](#) in the *Amazon Simple Storage Service Developer Guide*.

You can disable notifications by adding the empty `NotificationConfiguration` element.

By default, only the bucket owner can configure notifications on a bucket. However, bucket owners can use a bucket policy to grant permission to other users to set this configuration with `s3:PutBucketNotification` permission.

### Note

The PUT notification is an atomic operation. For example, suppose your notification configuration includes SNS topic, SQS queue, and Lambda function configurations. When you send a PUT request with this configuration, Amazon S3 sends test messages to your SNS topic. If the message fails, the entire PUT operation will fail, and Amazon S3 will not add the configuration to your bucket.

## Requests

### Syntax

```
PUT /?notification HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))

<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>
<QueueConfiguration>
    <Id>ConfigurationId</Id>
    <Filter>
        ...
    </Filter>
    <Queue>QueueARN</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
    ...
</QueueConfiguration>
...
<CloudFunctionConfiguration>
    <Id>ConfigurationId</Id>
    <Filter>
        ...
    </Filter>
    <CloudFunction>cloud-function-arn</CloudFunction>
    <Event>event-type</Event>
    ...
</CloudFunctionConfiguration>
...
</NotificationConfiguration>

```

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

Name	Description	Required
CloudFunction	<p>Lambda cloud function ARN that Amazon S3 can invoke when it detects events of the specified type.</p> <p>Type: String</p> <p>Ancestor: CloudFunctionConfiguration</p>	Required if CloudFunctionConfiguration is added.
CloudFunctionConfiguration	<p>Container for specifying the AWS Lambda notification configuration.</p> <p>Type: Container</p>	No

Name	Description	Required
	<p>Children: An <code>Id</code>, <code>Filter</code>, <code>CloudFunction</code>, and one, or more <code>Event</code>.</p> <p>Ancestor: <code>NotificationConfiguration</code></p>	
<code>Event</code>	<p>Bucket event for which to send notifications.</p> <p><b>Note</b> You can add multiple instance of <code>QueueConfiguration</code>, <code>TopicConfiguration</code>, or <code>CloudFunctionConfiguration</code> to the notification configuration.</p> <p>Type: String</p> <p>Valid Values: For a list of supported event types, go to <a href="#">Configuring Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Ancestor: <code>TopicConfiguration</code>, <code>QueueConfiguration</code>, and <code>CloudFunctionConfiguration</code>.</p>	Required if <code>TopicConfiguration</code> , <code>QueueConfiguration</code> or <code>CloudFunctionConfiguration</code> is added.
<code>Filter</code>	<p>Container for <code>S3Key</code>, which contains object key name filtering rules. For information about key name filtering, go to <a href="#">Configuring Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Container</p> <p>Children: <code>S3Key</code></p> <p>Ancestor: <code>TopicConfiguration</code>, <code>QueueConfiguration</code>, or <code>CloudFunctionConfiguration</code>.</p>	No
<code>FilterRule</code>	<p>Container for key value pair that defines the criteria for the filter rule.</p> <p>Container <code>S3Key</code></p> <p>Type: Container</p> <p>Children: <code>Name</code> and <code>Value</code></p> <p>Ancestor: <code>S3Key</code></p>	No
<code>Id</code>	<p>Optional unique identifier for each of the configurations in the <code>NotificationConfiguration</code>. If you don't provide, Amazon S3 will assign an ID.</p> <p>Type: String</p> <p>Ancestor: <code>TopicConfiguration</code> and <code>QueueConfiguration</code></p>	No

Name	Description	Required
Name	<p>Object key name prefix or suffix identifying one or more objects to which the filtering rule applies. Maximum prefix length can be up to 1,024 characters. Overlapping prefixes and suffixes are not supported. For more information, go to <a href="#">Configuring Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Ancestor: <code>FilterRule</code></p> <p>Valid values: prefix or suffix</p>	No
NotificationConfiguration	<p>Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.</p> <p>Type: Container</p> <p>Children: one or more <code>TopicConfiguration</code>, <code>QueueConfiguration</code>, and <code>CloudFunctionConfiguration</code> elements.</p> <p>Ancestor: None</p>	Yes
Queue	<p>Amazon SQS queue ARN to which Amazon S3 will publish a message when it detects events of specified type.</p> <p>Type: String</p> <p>Ancestor: <code>TopicConfiguration</code></p>	Required if <code>QueueConfiguration</code> is added.
QueueConfiguration	<p>Container for specifying the SQS queue configuration for the notification. You can add one or more of these queue configurations, each identifying one or more event types.</p> <p>Type: Container</p> <p>Children: An <code>Id</code>, <code>Filter</code>, <code>Topic</code>, and one, or more <code>Event</code>.</p> <p>Ancestor: <code>NotificationConfiguration</code></p>	No
S3Key	<p>Container for object key name prefix and suffix filtering rules.</p> <p>Type: Container</p> <p>Children: One or more <code>FilterRule</code></p> <p>Ancestor: <code>Filter</code></p>	No
Topic	<p>Amazon SNS topic ARN to which Amazon S3 will publish a message when it detects events of specified type.</p> <p>Type: String</p> <p>Ancestor: <code>TopicConfiguration</code></p>	Required if <code>TopicConfiguration</code> is added.

Name	Description	Required
TopicConfiguration	<p>Container for specifying an SNS topic configuration for the notification.</p> <p>Type: Container</p> <p>Children: An <code>Id</code>, <code>Filter</code>, <code>Topic</code>, and one, or more <code>Event</code>.</p> <p>Ancestor: <code>NotificationConfiguration</code></p>	No
Value	<p>Specifies the object key name prefix or suffix to filter on.</p> <p>Type: String</p> <p>Ancestor: <code>FilterRule</code></p>	No

## Responses

### Response Headers

In addition to the common response headers (see [Common Response Headers \(p. 4\)](#)), if the configuration in the request body includes only one `TopicConfiguration` specifying only the `s3:ReducedRedundancyLostObject` event type, the response will also include the `x-amz-sns-test-message-id` header containing the message ID of the test notification sent to topic.

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

Amazon S3 checks the validity of the proposed `NotificationConfiguration` element and verifies whether the proposed configuration is valid when you call the `PUT` operation. The following table lists the errors and possible causes.

HTTP Error	Code	Cause
HTTP 400 Bad Request	InvalidArgumentException	<p>The following conditions can cause this error:</p> <ul style="list-style-type: none"> <li>• A specified event is not supported for notifications.</li> <li>• A specified destination ARN does not exist or is not well-formed. Verify the destination ARN.</li> <li>• A specified destination is in a different region than the bucket. You must use a destination that resides in the same region as the bucket.</li> <li>• The bucket owner does not have appropriate permissions on the specified destination.</li> <li>• An object key name filtering rule defined with overlapping prefixes, overlapping suffixes, or overlapping combinations of prefixes and suffixes for the same event types.</li> </ul>

HTTP Error	Code	Cause
HTTP 403 Forbidden	AccessDenied	You are not the owner of the specified bucket, or you do not have the s3:PutBucketNotification bucket permission to set the notification configuration on the bucket.

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Configure Notification to Invoke a cloud function in Lambda

The following notification configuration includes CloudFunctionConfiguration, which identifies the event type for which Amazon S3 can invoke a cloud function and the name of the cloud function to invoke.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
    <Id>ObjectCreatedEvents</Id>
    <CloudFunction>arn:aws:lambda:us-west-2:35667example:function:CreateThumbnail</CloudFunction>
    <Event>s3:ObjectCreated:*</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

The following PUT uploads the notification configuration. The operation replaces the existing notification configuration.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 23:14:52 +0000
Content-Length: length

[request body]
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: 8+FlwagBSoT2qpMaGlfCUkRkFR5W3OeS7UhhoBb17j+kqvps2cSF1gJ5coLd53d2
x-amz-request-id: E5BA4600A3937335
Date: Fri, 31 Oct 2014 01:49:50 GMT
Content-Length: 0
Server: AmazonS3
```

### Example 2: Configure a Notification with Multiple Destinations

The following notification configuration includes the topic and queue configurations:

- A topic configuration identifying an SNS topic for Amazon S3 to publish events of the s3:ReducedRedundancyLostObject type.
- A queue configuration identifying an SQS queue for Amazon S3 to publish events of the s3:ObjectCreated:\* type.

```
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-east-1:356671443308:s3notificationtopic2</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
  </TopicConfiguration>
  <QueueConfiguration>
    <Queue>arn:aws:sqs:us-east-1:356671443308:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

The following PUT request against the notification subresource of the examplebucket bucket sends the preceding notification configuration in the request body. The operation replaces the existing notification configuration on the bucket.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 22:58:43 +0000
Content-Length: 391
Expect: 100-continue
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: SlvJLkfunoAGILZK3KqHSSUq4kwbudkrROmESoHOOpDacULy+cxRoR1Svrfooyvg2A
x-amz-request-id: BB1BA8E12D6A80B7
Date: Mon, 13 Oct 2014 22:58:44 GMT
Content-Length: 0
Server: AmazonS3
```

## Example 3: Configure a Notification with Object Key Name Filtering

The following notification configuration contains a queue configuration identifying an Amazon SQS queue for Amazon S3 to publish events to of the s3:ObjectCreated:Put type. The events will be published whenever an object that has a prefix of images/ and a .jpg suffix is PUT to a bucket. For more examples of notification configurations that use filtering, go to [Configuring Event Notifications](#) in the *Amazon Simple Storage Service Developer Guide*.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
      <S3Key>
        <FilterRule>
          <Name>prefix</Name>
          <Value>images/</Value>
```

```
</FilterRule>
<FilterRule>
    <Name>suffix</Name>
    <Value>.jpg</Value>
</FilterRule>
</S3Key>
</Filter>
<Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
<Event>s3:ObjectCreated:Put</Event>
</QueueConfiguration>
</NotificationConfiguration>
```

The following PUT request against the notification subresource of the `examplebucket` bucket sends the preceding notification configuration in the request body. The operation replaces the existing notification configuration on the bucket.

```
PUT http://s3.amazonaws.com/examplebucket?notification= HTTP/1.1
User-Agent: s3curl 2.0
Host: s3.amazonaws.com
Pragma: no-cache
Accept: */*
Proxy-Connection: Keep-Alive
Authorization: authorization string
Date: Mon, 13 Oct 2014 22:58:43 +0000
Content-Length: length
Expect: 100-continue
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: SlvJLkfunoAGILZK3KqHSSUq4kwbudkrROmESoHOpDacULy+cxRoR1Svrfovg2A
x-amz-request-id: BB1BA8E12D6A80B7
Date: Mon, 13 Oct 2014 22:58:44 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [GET Bucket notification \(p. 164\)](#)

# PUT Bucket object lock configuration

Service: Amazon Simple Storage Service

Places an Object Lock configuration on the specified bucket. The rule specified in the Object Lock configuration will be applied by default to every new object placed in the specified bucket.

## Request Syntax

```
PUT /?object-lock HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests \(AWS Signature Version 4\))
```

## URI Request Parameters

The request does not use any URI parameters.

## Request Body

The request accepts the following data in XML format.

### [ObjectLockConfiguration \(p. 298\)](#)

Root level tag for the ObjectLockConfiguration parameters.

Required: Yes

#### [ObjectLockEnabled \(p. 298\)](#)

Indicates whether this bucket has an Object Lock configuration enabled.

Type: String

Valid Values: Enabled

Required: Yes

#### [Rule \(p. 298\)](#)

The Object Lock rule that should be applied to objects placed in this bucket.

Type: [ObjectLockRule \(p. 332\)](#) object

Required: No

Example Request Body:

```
<ObjectLockConfiguration>
  <ObjectLockEnabled>Enabled</ObjectLockEnabled>
  <Rule>
    <DefaultRetention>
      <Mode>GOVERNANCE</Mode>
      <Days>30</Days>
    </DefaultRetention>
  </Rule>
</ObjectLockConfiguration>
```

## Response Syntax

HTTP/1.1 200

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

## Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

# PUT Bucket policy

## Description

This implementation of the `PUT` operation uses the `policy` subresource to return the policy of a specified bucket. If you are using an identity other than the root user of the AWS account that owns the bucket, the calling identity must have the `PutBucketPolicy` permissions on the specified bucket and belong to the bucket owner's account in order to use this operation.

If you don't have `PutBucketPolicy` permissions, Amazon S3 returns a `403 Access Denied` error. If you have the correct permissions, but you're not using an identity that belongs to the bucket owner's account, Amazon S3 returns a `405 Method Not Allowed` error.

### Important

As a security precaution, the root user of the AWS account that owns a bucket can always use this operation, even if the policy explicitly denies the root user the ability to perform this action.

For more information about bucket policies, see [Using Bucket Policies and User Policies](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Policy written in JSON
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

The body is a JSON string containing the policy contents containing the policy statements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

`PUT` response elements return whether the operation succeeded or not.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request shows the `PUT` individual policy request for the bucket.

```
PUT /?policy HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Tue, 04 Apr 2010 20:34:56 GMT
Authorization: authorization string

{
"Version":"2008-10-17",
"Id":"aaaa-bbbb-cccc-dddd",
"Statement" : [
    {
        "Effect":"Allow",
        "Sid":"1",
        "Principal" : {
            "AWS":["111122223333","444455556666"]
        },
        "Action":["s3:*"],
        "Resource":"arn:aws:s3:::bucket/*"
    }
]
}
```

### Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: Uuag1LuByR5Onimru9SAMPLEAtRPfTaOFG==
x-amz-request-id: 656c76696e6727732SAMPLE7374
Date: Tue, 04 Apr 2010 20:34:56 GMT
Connection: keep-alive
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 227\)](#)
- [DELETE Bucket \(p. 78\)](#)

# PUT Bucket replication

## Description

Creates a replication configuration or replaces one. For more information, see [Cross-Region Replication \(CRR\)](#) in the *Amazon S3 Developer Guide*.

## Requests

### Syntax

```
PUT /?replication HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5

Replication configuration XML in the body
```

For more information, see the following topics:

- For an overview of replication configuration XML and examples, see [Replication Configuration Overview](#) in the *Amazon S3 Developer Guide*.

#### Important

This topic describes all of the XML elements that are supported in the latest version of the replication configuration XML. For backward compatibility, Amazon S3 also continues to support earlier versions. For more information, see [Backward Compatibility](#) in the *Amazon S3 Developer Guide*.

- For authorization, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see <a href="#">RFC 1864</a> .  Type: String  Default: None	Yes

## Request Body

Specify the replication configuration in the request body. In the replication configuration, you provide the name of the destination bucket where you want Amazon S3 to replicate objects, the IAM role that Amazon S3 can assume to replicate objects on your behalf, and other relevant information.

A replication configuration must include at least one rule, and can contain a maximum of 1,000. Each rule identifies a subset of objects to replicate by filtering the objects in the source bucket. To choose additional subsets of objects to replicate, add a rule for each subset. All rules must specify the same destination bucket.

You can add other configuration options to rules. For more information, see [Replication Configuration Overview](#) in the *Amazon S3 Developer Guide*.

The following table describes the XML elements in a replication configuration.

Name	Description	Required
Account	<p>The account ID of the destination bucket owner. In a cross-account scenario, if you tell Amazon S3 to change replica ownership to the AWS account that owns the destination bucket by adding the <code>AccessControlTranslation</code> element, this is the account ID of the destination bucket owner. For more information, see <a href="#">Cross-Region Replication Additional Configuration: Change Replica Owner</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Ancestor: <code>Destination</code></p> <p>Container for replication rules. You can add a maximum of 1,000 rules. The maximum size of a replication configuration size is 2 MB.</p> <p>Type: Container</p> <p>Children: <code>Rule</code></p> <p>Ancestor: <code>None</code></p>	Yes, if you specify the <code>AccessControlTranslation</code> element
ReplicationConfiguration	<p>A container for replication rules. You can add a maximum of 1,000 rules. The maximum size of a replication configuration size is 2 MB.</p> <p>Type: Container</p> <p>Children: <code>Rule</code></p> <p>Ancestor: <code>None</code></p>	Yes
Role	<p>The Amazon Resource Name (ARN) of an IAM role that Amazon S3 can assume when replicating the objects.</p> <p>Type: String</p>	Yes

Name	Description	Required
	Ancestor: Rule	
Rule	<p>A container for information about a particular replication rule. A replication configuration must include at least one rule, and can contain up to 1,000 rules.</p> <p>Type: Container</p> <p>Ancestor:ReplicationConfiguration</p>	Yes
ID	<p>A unique identifier for the rule. The value is limited to 255 characters.</p> <p>Type: String</p> <p>Ancestor: Rule</p>	No
Status	<p>If you don't set the Status to Enabled, the rule is ignored.</p> <p>Type: String</p> <p>Ancestor: Rule</p> <p>Valid values: Enabled, Disabled</p>	Yes
Destination	<p>A container for destination information.</p> <p>Type: Container</p> <p>Ancestor: Rule</p>	Yes
Bucket	<p>The Amazon Resource Name (ARN) of the bucket where you want Amazon S3 to store replicas of the objects identified by the rule.</p> <p>If you have multiple rules, all rules must specify the same bucket as the destination. A replication configuration can replicate objects to only one destination bucket.</p> <p>Type: String</p> <p>Ancestor: Destination</p>	Yes

Name	Description	Required
StorageClass	<p>An optional destination storage class override to use when replicating objects. If you don't specify a storage class, Amazon S3 uses the storage class of the source object for object replicas.</p> <p>Type: String</p> <p>Ancestor: Destination</p> <p>Default: Storage class of the source object</p> <p>Valid values: STANDARD   STANDARD_IA   ONEZONE_IA   REDUCED_REDUNDANCY</p> <p>Constraints:</p> <ul style="list-style-type: none"> <li>You can't specify GLACIER as the storage class. To transition objects to the GLACIER storage class, use lifecycle configuration. For more information, see <a href="#">Object Lifecycle Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</li> <li>If you specify the STANDARD_IA or ONEZONE_IA storage class for object replicas, there are pricing considerations if the object replicas are less than 128 KB. For more information, see <a href="https://aws.amazon.com/s3/pricing/">https://aws.amazon.com/s3/pricing/</a>.</li> </ul>	No
AccessControlTranslation	<p>Use only in a cross-account scenario, where different AWS accounts own source and destination buckets, to change replica ownership to the AWS account that owns the destination bucket.</p> <p>If you don't add this element to the replication configuration, replicas are owned by same AWS account that owns the source object.</p> <p>Type: String</p> <p>Ancestor: Destination</p>	No
Owner	<p>Identifies the replica owner.</p> <p>Type: String</p> <p>Ancestor: AccessControlTranslation</p> <p>Default: Storage class of the source object</p> <p>Valid values: Destination</p>	Yes, if AccessControlTranslation is specified

## Specifying a Filter

To specify a subset of the objects in the source bucket to apply a replication rule to, add the `Filter` element as a child of the `Rule` element. You can filter objects based on an object key prefix, one or more object tags, or both. The following table describes the elements for filtering in a `Rule`.

Name	Description	Required
<code>Filter</code>	<p>A container that describes the filters that identify the source objects that you want to replicate.</p> <p>You can optionally specify one of these child elements: <code>Prefix</code>, <code>Tag</code>, or <code>And</code>.</p> <p>Use the <code>And</code> child element to specify an object filter that combines an object key <code>Prefix</code> and one or more <code>Tags</code>.</p> <p>An empty <code>Filter</code> element indicates that the rule applies to all objects.</p> <p>Ancestor: <code>Rule</code></p>	Yes.
<code>And</code>	<p>A container element for a <code>Prefix</code> and one or more <code>Tag</code> elements. At least one child element is required.</p> <p>Ancestor: <code>Filter</code></p>	Yes, if you want to specify more than one filtering criteria. For example, one object key prefix and one or more object tags.
<code>Prefix</code>	<p>An object key prefix that identifies one or more objects to which the rule applies. The maximum length of a <code>Prefix</code> is 1,024 characters. If prefixes in multiple rules overlap (if multiple rules apply to the same object), rule priority determines which rule applies to the object.</p> <p><b>Note</b>            In previous versions of replication configuration, only the object key prefix could be used as a rule filter (where you add the <code>Prefix</code> element as a child of the <code>Rule</code> element). Amazon S3 supports this for backward compatibility. But in the latest configuration, Amazon S3 allows you to specify either the <code>Filter</code> or <code>Prefix</code> as child of the <code>Rule</code>. For more information, see <a href="#">Backward Compatibility</a> in the <i>Amazon S3 Developer Guide</i>.</p> <p>Type: String</p> <p>Ancestor: <code>Filter</code></p>	No
<code>Tag</code>	<p>A container that provides a tag key and value.</p> <p>Ancestor: <code>Filter</code></p>	No

Name	Description	Required
Key	<p>Provides an object tag key. The Tag Key and Value are case sensitive. A Tag Key can have 1-128 characters.</p> <p>Type: String</p> <p>Ancestor: <code>EncryptionConfiguration</code></p>	No
Value	<p>Provides the object Tag Value. The Tag Key and Value are case sensitive. The Tag Value can have 0-256 characters.</p> <p>Type: String</p> <p>Ancestor: <code>EncryptionConfiguration</code></p>	No

When you add the `Filter` element in the configuration, you must also add the elements described in this table.

Name	Description	Required
<code>DeleteMarkerReplication</code>	<p>A container that describes whether Amazon S3 replicates the delete markers. If you specify a <code>Filter</code>, you must specify this element. However, in the latest version of replication configuration (when <code>Filter</code> is specified), Amazon S3 doesn't replicate delete markers. Therefore, the <code>DeleteMarkerReplication</code> element can contain only <code>&lt;Status&gt;Disabled&lt;/Status&gt;</code>. For an example configuration, see <a href="#">The Basic Rule Configuration</a> in the <i>Amazon S3 Developer Guide</i>.</p> <p><b>Note</b> If you don't specify the <code>Filter</code> element, Amazon S3 assumes the replication configuration is the earlier version, V1. In the earlier version, Amazon S3 handled replication of delete markers differently. For more information, see <a href="#">Backward Compatibility</a> in the <i>Amazon S3 Developer Guide</i>.</p> <p>Ancestor: <code>Rule</code></p>	Yes, if <code>Filter</code> is specified
<code>Status</code>	<p>Indicates whether to replicate delete markers.</p> <p>Type: String</p> <p>Ancestor: <code>DeleteMarkerReplication</code></p> <p>Valid values: <code>Disabled</code></p>	Yes, if <code>DeleteMarkerReplication</code> is specified
<code>Priority</code>	If you specify multiple rules with overlapping filters, identifies the rule priority. For example, if two rules apply to the same object based on the <code>Filter</code> specified, then the rule with higher	Yes, if <code>Filter</code> is specified

Name	Description	Required
	<p>priority supersedes. The higher the numerical value of this element, the higher the rule priority. For more information, see <a href="#">Backward Compatibility</a> in the <i>Amazon S3 Developer Guide</i>.</p> <p>Type: Integer</p> <p>Ancestor: Rule</p> <p>Valid values: 0 - INT-MAX.</p>	

## Handling Replication of Encrypted Objects

By default, Amazon S3 doesn't replicate objects that are stored at rest using server-side encryption with AWS KMS-managed keys. To replicate AWS KMS-encrypted objects, add the following optional configuration. For information about replication configuration, see [CRR: Replicating Objects Created with SSE Using AWS KMS-Managed Encryption Keys](#) in the *Amazon Simple Storage Service Developer Guide*.

Name	Description	Required
SourceSelectionCriteria	<p>A container that describes additional filters that identify the source objects that you want to replicate.</p> <p>Currently, Amazon S3 supports only the filter for objects created with server-side encryption using an AWS KMS-managed key. You can choose to enable or disable replication of these objects.</p> <p>Ancestor: Rule</p>	Yes, if you want Amazon S3 to replicate objects created with server-side encryption using AWS KMS-managed keys
SseKmsEncryptedObjects	<p>A container element for Status.</p> <p>Ancestor: SourceSelectionCriteria</p>	Yes, if SourceSelectionCriteria is specified
Status	<p>A flag that tells Amazon S3 whether to replicate objects created with server-side encryption using an AWS KMS-managed key.</p> <p>Type: String</p> <p>Ancestor: SseKmsEncryptedObjects</p> <p>Valid Values: Enabled, Disabled</p>	Yes, if SseKmsEncryptedObjects is specified
EncryptionConfiguration	<p>A container that provides encryption-related information.</p> <p>Ancestor: Destination</p>	Yes, if SourceSelectionCriteria is specified
ReplicaKmsKeyID	Provides the AWS KMS Key ID (Key ARN or Alias ARN) of the destination bucket. Amazon S3 uses this key to encrypt replicas.	Yes, if EncryptionConfiguration is specified

Name	Description	Required
	Type: String  Ancestor: EncryptionConfiguration	

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

When you call the `PUT` operation, Amazon S3 checks the validity of the proposed `AnalyticsConfiguration` element and verifies that the proposed configuration is valid. The following table lists errors and possible causes.

HTTP Error	Code	Cause
HTTP 400	InvalidRequest	If the <code>&lt;Owner&gt;</code> in <code>&lt;AccessControlTranslation&gt;</code> has a value, the <code>&lt;Account&gt;</code> element must be specified.
HTTP 400	InvalidArgumentException	The <code>&lt;Account&gt;</code> element is empty. It must contain a valid account ID.
HTTP 400	InvalidArgumentException	The AWS account specified in the <code>&lt;Account&gt;</code> element must match the destination bucket owner.

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

The following example shows how to add a replication configuration.

### Example 1: Add a Replication Configuration

The following is a sample `PUT` request that creates a `replication` subresource on the specified bucket and saves the replication configuration in it. The replication configuration specifies a rule to replicate objects to the `exampletargetbucket` bucket. The rule includes a filter to replicate only the objects created with the key name prefix `TaxDocs` and that have two specific tags.

After you add a replication configuration to your bucket, Amazon S3 assumes the AWS Identity and Access Management (IAM) role specified in the configuration to replicate objects on behalf of the bucket owner. The bucket owner is the AWS account that created the bucket.

```
PUT /?replication HTTP/1.1
```

```
Host: examplebucket.s3.amazonaws.com
Date: Wed, 11 Feb 2015 02:11:21 GMT
Content-MD5: q6yJDlIkBaGGfb3QLY69A==
Authorization: authorization string
Content-Length: length

<ReplicationConfiguration>
  <Role>arn:aws:iam::35667example:role/CrossRegionReplicationRoleForS3</Role>
  <Rule>
    <ID>rule1</ID>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
      <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
      <And>
        <Prefix>TaxDocs</Prefix>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
        <Tag>
          <Key>key1</Key>
          <Value>value1</Value>
        </Tag>
      </And>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::exampletargetbucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

The following is a sample response:

```
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJ0JJYcd+1j5nM/rUFiiiz/fNbDOsd3JUE8NWMLNHXmvPfwMpdc
x-amz-request-id: 9E26D08072A8EF9E
Date: Wed, 11 Feb 2015 02:11:22 GMT
Content-Length: 0
Server: AmazonS3
```

Filtering using the `<Filter>` element is supported in the latest XML configuration. If you are using an earlier version of the XML configuration, you can filter only on key prefix. In that case, you add the `<Prefix>` element as a child of the `<Rule>`.

For more examples of replication configuration, see [Replication Configuration Overview](#) in the *Amazon S3 Developer Guide*.

## Related Resources

- [GET Bucket replication \(p. 187\)](#).
- [DELETE Bucket replication \(p. 95\)](#).
- For information about enabling versioning on a bucket, see [Using Versioning](#) in the *Amazon Simple Storage Service Developer Guide*.
- By default, a resource owner, in this case the AWS account that created the bucket, can perform this operation. The resource owner can also grant others permissions to perform the operation. For more information, see the following topics in the *Amazon Simple Storage Service Developer Guide*:
  - [Specifying Permissions in a Policy](#)

- [Managing Access Permissions to Your Amazon S3 Resources](#)

# PUT Bucket requestPayment

## Description

This implementation of the `PUT` operation uses the `requestPayment` subresource to set the request payment configuration of a bucket. By default, the bucket owner pays for downloads from the bucket. This configuration parameter enables the bucket owner (only) to specify that the person requesting the download will be charged for the download. For more information, see [Requester Pays Buckets](#).

## Requests

### Syntax

```
PUT ?requestPayment HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization:signatureValue

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>payer</Payer>
</RequestPaymentConfiguration>
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

Name	Description
<code>Payer</code>	Specifies who pays for the download and request fees.  Type: Enum  Valid Values: Requester   BucketOwner  Ancestor: RequestPaymentConfiguration
<code>RequestPaymentConfiguration</code>	Container for <code>Payer</code> .  Type: Container

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This request creates a Requester Pays bucket named "colorpictures."

```
PUT ?requestPayment HTTP/1.1
Host: colorpictures.s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzSD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 227\)](#)
- [GET Bucket requestPayment \(p. 194\)](#)

# PUT Bucket tagging

## Description

This implementation of the `PUT` operation uses the `tagging` subresource to add a set of tags to an existing bucket.

Use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, to see the cost of combined resources, organize your billing information according to resources with the same tag key values. For example, you can tag several resources with a specific application name, and then organize your billing information to see the total cost of that application across several services. For more information, see [Cost Allocation and Tagging](#) in [About AWS Billing and Cost Management](#).

To use this operation, you must have permissions to perform the `s3:PutBucketTagging` action. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

The following request shows the syntax for sending tagging information in the request body.

```
PUT /?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))

<Tagging>
  <TagSet>
    <Tag>
      <Key>Tag Name</Key>
      <Value>Tag Value</Value>
    </Tag>
  </TagSet>
</Tagging>
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

Content-MD5 will be a required header for this operation.

### Request Elements

Name	Description	Required
Tagging	Container for the TagSet and Tag elements.  Type: String	Yes

Name	Description	Required
	Ancestors: None	
TagSet	Container for a set of tags  Type: Container  Ancestors: Tagging	Yes
Tag	Container for tag information.  Type: Container  Ancestors: TagSet	Yes
Key	Name of the tag.  Type: String  Ancestors: Tag	Yes
Value	Value of the tag.  Type: String  Ancestors: Tag	Yes

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not return response elements.

### Special Errors

- **InvalidTagError** - The tag provided was not a valid tag. This error can occur if the tag did not pass input validation. For information about tag restrictions, see [User-Defined Tag Restrictions](#) and [AWS-Generated Cost Allocation Tag Restrictions](#) in the *AWS Billing and Cost Management User Guide*.
- **MalformedXMLError** - The XML provided does not match the schema.
- **OperationAbortedError** - A conflicting conditional operation is currently in progress against this resource. Please try again.
- **InternalError** - The service was unable to apply the provided tag to the bucket.

## Examples

### Sample Request: Add tag set to a bucket

The following request adds a tag set to the existing `examplebucket` bucket.

```
PUT ?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2012 20:04:21 GMT
Authorization: authorization string

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>jsmith</Value>
    </Tag>
  </TagSet>
</Tagging>
```

## Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Oct 2012 12:00:00 GMT
```

## Related Resources

- [GET Bucket tagging \(p. 196\)](#)
- [DELETE Bucket tagging \(p. 97\)](#)

# PUT Bucket versioning

## Description

This implementation of the `PUT` operation uses the `versioning` subresource to set the versioning state of an existing bucket. To set the versioning state, you must be the bucket owner.

You can set the versioning state with one of the following values:

- **Enabled**—Enables versioning for the objects in the bucket
  - All objects added to the bucket receive a unique version ID.
- **Suspended**—Disables versioning for the objects in the bucket
  - All objects added to the bucket receive the version ID `null`.

If the versioning state has never been set on a bucket, it has no versioning state; a `GET versioning` request does not return a versioning state value.

If the bucket owner enables MFA Delete in the bucket versioning configuration, the bucket owner must include the `x-amz-mfa` request header and the `Status` and the `MfaDelete` request elements in a request to set the versioning state of the bucket.

### Important

If you have an object expiration lifecycle policy in your non-versioned bucket and you want to maintain the same permanent delete behavior when you enable versioning, you must add a noncurrent expiration policy. The noncurrent expiration lifecycle policy will manage the deletes of the noncurrent object versions in the version-enabled bucket. (A version-enabled bucket maintains one current and zero or more noncurrent object versions.) For more information, see [Lifecycle and Versioning](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information about creating a bucket, see [PUT Bucket \(p. 227\)](#). For more information about returning the versioning state of a bucket, see [GET Bucket Versioning Status \(p. 199\)](#).

## Requests

### Syntax

```
PUT /?versioning HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
x-amz-mfa: [SerialNumber] [TokenCode]

<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Note the space between `[SerialNumber]` and `[TokenCode]`.

### Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

Name	Description	Required
x-amz-mfa	<p>The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device.</p> <p>Type: String</p> <p>Default: None</p> <p>Condition: Required to configure the versioning state if versioning is configured with MFA Delete enabled.</p>	Conditional

## Request Elements

Name	Description	Required
Status	<p>Sets the versioning state of the bucket.</p> <p>Type: Enum</p> <p>Valid Values: Suspended   Enabled</p> <p>Ancestor: VersioningConfiguration</p>	No
MfaDelete	<p>Specifies whether MFA Delete is enabled in the bucket versioning configuration. When enabled, the bucket owner must include the <code>x-amz-mfa</code> request header in requests to change the versioning state of a bucket and to permanently delete a versioned object.</p> <p>Type: Enum</p> <p>Valid Values: Disabled   Enabled</p> <p>Ancestor: VersioningConfiguration</p> <p>Constraint: Can only be used when you use Status.</p>	No
VersioningConfiguration	<p>Container for setting the versioning state.</p> <p>Type: Container</p> <p>Children: Status</p> <p>Ancestor: None</p>	Yes

# Responses

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

# Examples

## Sample Request

The following request enables versioning for the specified bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
```

## Sample Request

The following request suspends versioning for the specified bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Suspended</Status>
</VersioningConfiguration>
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT
```

## Sample Request

The following request enables versioning and MFA Delete on a bucket.

```
PUT /?versioning HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-mfa:[SerialNumber] [TokenCode]
Authorization: authorization string
Content-Type: text/plain
Content-Length: 124

<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
  <MfaDelete>Enabled</MfaDelete>
</VersioningConfiguration>
```

Note the space between [SerialNumber] and [TokenCode] and that you must include Status whenever you use MfaDelete.

## Sample Response

```
HTTPS/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2006 12:00:00 GMT

Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [DELETE Bucket \(p. 78\)](#)
- [PUT Bucket \(p. 227\)](#)

# PUT Bucket website

## Description

Sets the configuration of the website that is specified in the `website` subresource. To configure a bucket as a website, you can add this subresource on the bucket with website configuration information such as the file name of the index document and any redirect rules. For more information, go to [Hosting Websites on Amazon S3](#) in the *Amazon Simple Storage Service Developer Guide*.

This `PUT` operation requires the `S3:PutBucketWebsite` permission. By default, only the bucket owner can configure the website attached to a bucket; however, bucket owners can allow other users to set the website configuration by writing a bucket policy that grants them the `S3:PutBucketWebsite` permission.

## Requests

### Syntax

```
PUT /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Content-Length: ContentLength
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/"> <!-- website configuration information. --&gt;
&lt;/WebsiteConfiguration&gt;</pre>
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

You can use a website configuration to redirect all requests to the website endpoint of a bucket, or you can add routing rules that redirect only specific requests.

- To redirect all website requests sent to the bucket's website endpoint, you add a website configuration with the following elements. Because all requests are sent to another website, you don't need to provide index document name for the bucket.

Name	Description	Required
WebsiteConfiguration	The root element for the website configuration  Type: Container	Yes

Name	Description	Required
	Ancestors: None	
RedirectAllRequestsTo	<p>Describes the redirect behavior for every request to this bucket's website endpoint. If this element is present, no other siblings are allowed.</p> <p>Type: Container</p> <p>Ancestors: WebsiteConfiguration</p>	Yes
HostName	<p>Name of the host where requests will be redirected.</p> <p>Type: String</p> <p>Ancestors: RedirectAllRequestsTo</p>	Yes
Protocol	<p>Protocol to use (http, https) when redirecting requests. The default is the protocol that is used in the original request.</p> <p>Type: String</p> <p>Ancestors: RedirectAllRequestsTo</p>	No

- If you want granular control over redirects, you can use the following elements to add routing rules that describe conditions for redirecting requests and information about the redirect destination. In this case, the website configuration must provide an index document for the bucket, because some requests might not be redirected.

Name	Description	Required
WebsiteConfiguration	<p>Container for the request</p> <p>Type: Container</p> <p>Ancestors: None</p>	Yes
IndexDocument	<p>Container for the Suffix element.</p> <p>Type: Container</p> <p>Ancestors: WebsiteConfiguration</p>	Yes
Suffix	<p>A suffix that is appended to a request that is for a <i>directory</i> on the website endpoint (e.g., if the suffix is index.html and you make a request to samplebucket/images/, the data that is returned will be for the object with the key name images/index.html)</p> <p>The suffix must not be empty and must not include a slash character.</p> <p>Type: String</p> <p>Ancestors: WebsiteConfiguration.IndexDocument</p>	Yes
ErrorDocument	Container for the Key element	No

Name	Description	Required
	Type: Container  Ancestors: WebsiteConfiguration	
Key	The object key name to use when a 4XX class error occurs. This key identifies the page that is returned when such an error occurs.  Type: String  Ancestors: WebsiteConfiguration.ErrorDocument  Condition: Required when ErrorDocument is specified.	Conditional
RoutingRules	Container for a collection of RoutingRule elements.  Type: Container  Ancestors: WebsiteConfiguration	No
RoutingRule	Container for one routing rule that identifies a condition and a redirect that applies when the condition is met.  Type: String  Ancestors: WebsiteConfiguration.RoutingRules  Condition: In a RoutingRules container, there must be at least one of RoutingRule element.	Yes
Condition	A container for describing a condition that must be met for the specified redirect to apply. For example: <ul style="list-style-type: none"><li>• If request is for pages in the /docs folder, redirect to the /documents folder.</li><li>• If request results in HTTP error 4xx, redirect request to another host where you might process the error.</li></ul> Type: Container  Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule	No

Name	Description	Required
KeyPrefixEquals	<p>The object key name prefix when the redirect is applied. For example, to redirect requests for <code>ExamplePage.html</code>, the key prefix will be <code>ExamplePage.html</code>. To redirect request for all pages with the prefix <code>docs/</code>, the key prefix will be <code>/docs</code>, which identifies all objects in the <code>docs/</code> folder.</p> <p>Type: String</p> <p>Ancestors: <code>WebsiteConfiguration.RoutingRules.RoutingRule.Condition</code></p> <p>Condition: Required when the parent element <code>Condition</code> is specified and sibling <code>HttpErrorCodeReturnedEquals</code> is not specified. If both conditions are specified, both must be true for the redirect to be applied.</p>	Conditional
HttpErrorCodeReturnedEquals	<p>The HTTP error code when the redirect is applied. In the event of an error, if the error code equals this value, then the specified redirect is applied.</p> <p>Type: String</p> <p>Ancestors: <code>WebsiteConfiguration.RoutingRules.RoutingRule.Condition</code></p> <p>Condition: Required when parent element <code>Condition</code> is specified and sibling <code>KeyPrefixEquals</code> is not specified. If both are specified, then both must be true for the redirect to be applied.</p>	Conditional
Redirect	<p>Container for redirect information. You can redirect requests to another host, to another page, or with another protocol. In the event of an error, you can specify a different error code to return.</p> <p>Type: String</p> <p>Ancestors: <code>WebsiteConfiguration.RoutingRules.RoutingRule</code></p>	Yes
Protocol	<p>The protocol to use in the redirect request.</p> <p>Type: String</p> <p>Ancestors: <code>WebsiteConfiguration.RoutingRules.RoutingRule.Redirect</code></p> <p>Valid Values: <code>http</code>, <code>https</code></p> <p>Condition: Not required if one of the siblings is present</p>	No

Name	Description	Required
HostName	<p>The host name to use in the redirect request.</p> <p>Type: String</p> <p>Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Redirect</p> <p>Condition: Not required if one of the siblings is present</p>	No
ReplaceKeyPrefixWith	<p>The object key prefix to use in the redirect request. For example, to redirect requests for all pages with prefix docs/ (objects in the docs/ folder) to documents/, you can set a condition block with KeyPrefixEquals set to docs/ and in the Redirect set ReplaceKeyPrefixWith to /documents.</p> <p>Type: String</p> <p>Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Redirect</p> <p>Condition: Not required if one of the siblings is present. Can be present only if ReplaceKeyWith is not provided.</p>	No
ReplaceKeyWith	<p>The specific object key to use in the redirect request. For example, redirect request to error.html.</p> <p>Type: String</p> <p>Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Redirect</p> <p>Condition: Not required if one of the sibling is present. Can be present only if ReplaceKeyPrefixWith is not provided.</p>	No
HttpRedirectCode	<p>The HTTP redirect code to use on the response.</p> <p>Type: String</p> <p>Ancestors: WebsiteConfiguration.RoutingRules.RoutingRule.Redirect</p> <p>Condition: Not required if one of the siblings is present.</p>	No

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Examples

### Example 1: Configure bucket as a website (add website configuration)

The following request configures a bucket `example.com` as a website. The configuration in the request specifies `index.html` as the index document. It also specifies the optional error document, `SomeErrorDocument.html`.

```
PUT ?website HTTP/1.1
Host: example.com.s3.amazonaws.com
Content-Length: 256
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
    <IndexDocument>
        <Suffix>index.html</Suffix>
    </IndexDocument>
    <ErrorDocument>
        <Key>SomeErrorDocument.html</Key>
    </ErrorDocument>
</WebsiteConfiguration>
```

Amazon S3 returns the following sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 80CD4368BD211111
Date: Thu, 27 Jan 2011 00:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

### Example 2: Configure bucket as a website but redirect all requests

The following request configures a bucket `www.example.com` as a website; however, the configuration specifies that all GET requests for the `www.example.com` bucket's website endpoint will be redirected to host `example.com`.

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: length-value
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
    <RedirectAllRequestsTo>
        <HostName>example.com</HostName>
    </RedirectAllRequestsTo>
</WebsiteConfiguration>
```

This redirect can be useful when you want to serve requests for both `http://www.example.com` and `http://example.com`, but you want to maintain the website content in only one bucket, in this case `example.com`. For more information, go to [Hosting Websites on Amazon S3](#) in the *Amazon Simple Storage Service Developer Guide*.

## Example 3: Configure bucket as a website and also specify optional redirection rules

Example 1 is the simplest website configuration. It configures a bucket as a website by providing only an index document and an error document. You can further customize the website configuration by adding routing rules that redirect requests for one or more objects. For example, suppose your bucket contained the following objects:

```
index.html  
  
docs/article1.html  
  
docs/article2.html
```

If you decided to rename the folder from `docs/` to `documents/`, you would need to redirect requests for prefix `/docs` to `documents/`. For example, a request for `docs/article1.html` will need to be redirected to `documents/article1.html`.

In this case, you update the website configuration and add a routing rule as shown in the following request:

```
PUT ?website HTTP/1.1  
Host: www.example.com.s3.amazonaws.com  
Content-Length: length-value  
Date: Thu, 27 Jan 2011 12:00:00 GMT  
Authorization: signatureValue  
  
<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>  
  <IndexDocument>  
    <Suffix>index.html</Suffix>  
  </IndexDocument>  
  <ErrorDocument>  
    <Key>Error.html</Key>  
  </ErrorDocument>  
  
  <RoutingRules>  
    <RoutingRule>  
      <Condition>  
        <KeyPrefixEquals>docs/</KeyPrefixEquals>  
      </Condition>  
      <Redirect>  
        <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>  
      </Redirect>  
    </RoutingRule>  
  </RoutingRules>  
</WebsiteConfiguration>
```

## Example 4: Configure bucket as a website and redirect errors

You can use a routing rule to specify a condition that checks for a specific HTTP error code. When a page request results in this error, you can optionally reroute requests. For example, you might route requests to another host and optionally process the error. The routing rule in the following requests redirects requests to an EC2 instance in the event of an HTTP error 404. For illustration, the redirect also inserts a object key prefix `report-404/` in the redirect. For example, if you request a page `ExamplePage.html` and it results in a HTTP 404 error, the request is routed to a page `report-404/testPage.html` on the specified EC2 instance. If there is no routing rule and the HTTP error 404 occurred, then `Error.html` would be returned.

```
PUT ?website HTTP/1.1
```

```
Host: www.example.com.s3.amazonaws.com
Content-Length: 580
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>

  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
      </Condition>
      <Redirect>
        <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
        <ReplaceKeyPrefixWith>report-404/<ReplaceKeyPrefixWith>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

## Example 5: Configure a bucket as a website and redirect folder requests to a page

Suppose you have the following pages in your bucket:

```
images/photo1.jpg
images/photo2.jpg
images/photo3.jpg
```

Now you want to route requests for all pages with the `images/` prefix to go to a single page, `errorpage.html`. You can add a website configuration to your bucket with the routing rule shown in the following request:

```
PUT ?website HTTP/1.1
Host: www.example.com.s3.amazonaws.com
Content-Length: 481
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: signatureValue

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>

  <RoutingRules>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>images/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <ReplaceKeyWith>errorpage.html</ReplaceKeyWith>
```

```
</Redirect>
</RoutingRule>
</RoutingRules>
</WebsiteConfiguration>
```

# DefaultRetention

Service: Amazon Simple Storage Service

The container element for specifying the default Object Lock retention settings for new objects placed in the specified bucket.

## Contents

### Mode

The default Object Lock retention mode you want to apply to new objects placed in the specified bucket.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: Yes

### Days

The number of days that you want to specify for the default retention period.

Type: Integer

Required: No

### Years

The number of years that you want to specify for the default retention period.

Type: Integer

Required: No

### Note

Either Days or Years must be specified, but not both.

# ObjectLockConfiguration

Service: Amazon Simple Storage Service

The container element for Object Lock configuration parameters.

## Contents

### ObjectLockEnabled

Indicates whether this bucket has an Object Lock configuration enabled.

Type: String

Valid Values: Enabled

Required: Yes

### Rule

The Object Lock rule in place for the specified bucket.

Type: [ObjectLockRule \(p. 332\)](#) object

Required: No

# ObjectLockRule

Service: Amazon Simple Storage Service

The container element for an Object Lock rule.

## Contents

### **DefaultRetention**

The default retention period that you want to apply to new objects placed in the specified bucket.

Type: [DefaultRetention \(p. 330\)](#) object

Required: No

# Operations on Objects

This section describes operations you can perform on Amazon S3 objects.

## Topics

- [Delete Multiple Objects \(p. 333\)](#)
- [DELETE Object \(p. 343\)](#)
- [DELETE Object tagging \(p. 347\)](#)
- [GET Object \(p. 349\)](#)
- [GET Object ACL \(p. 361\)](#)
- [GET Object legal hold \(p. 365\)](#)
- [GET Object retention \(p. 366\)](#)
- [GET Object tagging \(p. 368\)](#)
- [GET Object torrent \(p. 371\)](#)
- [HEAD Object \(p. 373\)](#)
- [OPTIONS object \(p. 382\)](#)
- [POST Object \(p. 385\)](#)
- [POST Object restore \(p. 397\)](#)
- [PUT Object \(p. 412\)](#)
- [PUT Object legal hold \(p. 427\)](#)
- [PUT Object retention \(p. 429\)](#)
- [PUT Object - Copy \(p. 431\)](#)
- [PUT Object acl \(p. 447\)](#)
- [PUT Object tagging \(p. 454\)](#)
- [SELECT Object Content \(p. 457\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [Initiate Multipart Upload \(p. 492\)](#)
- [List Parts \(p. 502\)](#)
- [Upload Part \(p. 508\)](#)
- [Upload Part - Copy \(p. 514\)](#)
- [ObjectLockLegalHold \(p. 522\)](#)
- [ObjectLockRetention \(p. 523\)](#)

## Delete Multiple Objects

### Description

The Multi-Object Delete operation enables you to delete multiple objects from a bucket using a single HTTP request. If you know the object keys that you want to delete, then this operation provides a suitable alternative to sending individual delete requests (see [DELETE Object \(p. 343\)](#)), reducing per-request overhead.

The Multi-Object Delete request contains a list of up to 1000 keys that you want to delete. In the XML, you provide the object key names, and optionally, version IDs if you want to delete a specific version of the object from a versioning-enabled bucket. For each key, Amazon S3 performs a delete operation and

returns the result of that delete, success, or failure, in the response. Note that, if the object specified in the request is not found, Amazon S3 returns the result as deleted.

The Multi-Object Delete operation supports two modes for the response; verbose and quiet. By default, the operation uses verbose mode in which the response includes the result of deletion of each key in your request. In quiet mode the response includes only keys where the delete operation encountered an error. For a successful deletion, the operation does not return any information about the delete in the response body.

When performing a Multi-Object Delete operation on an MFA Delete enabled bucket, that attempts to delete any versioned objects, you must include an MFA token. If you do not provide one, the entire request will fail, even if there are non versioned objects you are attempting to delete. If you provide an invalid token, whether there are versioned keys in the request or not, the entire Multi-Object Delete request will fail. For information about MFA Delete, see [MFA Delete](#).

Finally, the Content-MD5 header is required for all Multi-Object Delete requests. Amazon S3 uses the header value to ensure that your request body has not been altered in transit.

## Requests

### Syntax

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Authorization: authorization string
Content-Length: Size
Content-MD5: MD5

<?xml version="1.0" encoding="UTF-8"?>
<Delete>
    <Quiet>true</Quiet>
    <Object>
        <Key>Key</Key>
        <VersionId>VersionId</VersionId>
    </Object>
    <Object>
        <Key>Key</Key>
    </Object>
    ...
</Delete>
```

### Request Parameters

The Multi-Object Delete operation requires a single query string parameter called "delete" to distinguish it from other bucket POST operations.

### Request Headers

This operation uses the following Request Headers in addition to the request headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the data. This header must be used as a message integrity check to verify that the request body was not corrupted in transit. For more information, go to <a href="#">RFC 1864</a> .  Type: String	Yes

Name	Description	Required
	Default: None	
Content-Length	Length of the body according to RFC 2616. Type: String Default: None	Yes
x-amz-mfa	The value is the concatenation of the authentication device's serial number, a space, and the value that is displayed on your authentication device. Type: String Default: None Condition: Required to permanently delete a versioned object if versioning is configured with MFA Delete enabled.	Conditional

## Request Elements

Name	Description	Required
Delete	Container for the request.  Ancestor: None  Type: Container  Children: One or more Object elements and an optional Quiet element.	Yes
Quiet	Element to enable quiet mode for the request. When you add this element, you must set its value to true.  Ancestor: Delete  Type: Boolean  Default: false	No
Object	Container element that describes the delete request for an object.  Ancestor: Delete  Type: Container  Children: Key element and an optional VersionId element.	Yes
Key	Key name of the object to delete.  Ancestor: Object  Type: String	Yes
VersionId	VersionId for the specific version of the object to delete.	No

Name	Description	Required
	Ancestor: Object  Type: String	

## Responses

### Response Headers

This operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
DeleteResult	Container for the response.  Children: Deleted, Error  Type: Container  Ancestor: None
Deleted	Container element for a successful delete. It identifies the object that was successfully deleted.  Children: Key, VersionId  Type: Container  Ancestor: DeleteResult
Key	Key name for the object that Amazon S3 attempted to delete.  Type: String  Ancestor: Deleted, or Error
VersionId	VersionId for the versioned object in the case of a versioned delete.  Type: String  Ancestor: Deleted
DeleteMarker	DeleteMarker element with a true value indicates that the request accessed a delete marker.  If a specific delete request either creates or deletes a delete marker, Amazon S3 returns this element in the response with a value of true. This is only the case when your Multi-Object Delete request is on a bucket that has versioning enabled or suspended. For more information about delete markers, go to <a href="#">Object Versioning</a> .

Name	Description
	<p>Type: Boolean</p> <p>Ancestor: Deleted</p>
DeleteMarkerVersionId	<p>Version ID of the delete marker accessed (deleted or created) by the request.</p> <p>If the specific delete request in the Multi-Object Delete either creates or deletes a delete marker, Amazon S3 returns this element in response with the version ID of the delete marker. When deleting an object in a bucket with versioning enabled, this value is present for the following two reasons:</p> <ul style="list-style-type: none"> <li>• You send a non-versioned delete request, that is, you specify only object key and not the version ID. In this case, Amazon S3 creates a delete marker and returns its version ID in the response.</li> <li>• You send a versioned delete request, that is, you specify an object key and a version ID in your request; however, the version ID identifies a delete marker. In this case, Amazon S3 deletes the delete marker and returns the specific version ID in response. For information about versioning, go to <a href="#">Object Versioning</a>.</li> </ul> <p>Type: String</p> <p>Ancestor: Deleted</p>
Error	<p>Container for a failed delete operation that describes the object that Amazon S3 attempted to delete and the error it encountered.</p> <p>Children: Key, VersionId, Code, Message.</p> <p>Type: String</p> <p>Ancestor: DeleteResult</p>
Key	<p>Key for the object Amazon S3 attempted to delete.</p> <p>Type: String</p> <p>Ancestor: Error</p>
VersionId	<p>Version ID of the versioned object Amazon S3 attempted to delete. Amazon S3 includes this element only in case of a versioned-delete request.</p> <p>Type: String</p> <p>Ancestor: Deleted, Error</p>

Name	Description
Code	Status code for the result of the failed delete. Type: <code>String</code> Values: <code>AccessDenied</code> , <code>InternalError</code> Ancestor: <code>Error</code>
Message	Error description. Type: <code>String</code> Ancestor: <code>Error</code>

## Examples

### Example 1: Multi-Object Delete resulting in mixed success/error response

This example illustrates a Multi-Object Delete request to delete objects that result in mixed success and errors response.

#### Sample Request

The following Multi-Object Delete request deletes two objects from a bucket (bucketname). In this example, the requester does not have permission to delete the sample2.txt object.

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIee=
Content-Length: 125
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>sample1.txt</Key>
  </Object>
  <Object>
    <Key>sample2.txt</Key>
  </Object>
</Delete>
```

#### Sample Response

The response includes a `DeleteResult` element that includes a `Deleted` element for the item that Amazon S3 successfully deleted and an `Error` element that Amazon S3 did not delete because you didn't have permission to delete the object.

```
HTTP/1.1 200 OK
x-amz-id-2: 5h4FxSNCUS7wP5z92eGCWDshNpMnRuXvETA4HH3LvvH6VAIr0jU7tH9kM7X+njXx
x-amz-request-id: A437B3B641629AEE
Date: Fri, 02 Dec 2011 01:53:42 GMT
```

```
Content-Type: application/xml
Server: AmazonS3
Content-Length: 251

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Deleted>
    <Key>sample1.txt</Key>
  </Deleted>
  <Error>
    <Key>sample2.txt</Key>
    <Code>AccessDenied</Code>
    <Message>Access Denied</Message>
  </Error>
</DeleteResult>
```

## Example 2: Deleting Object from a Versioned Bucket

If you delete an item from a versioning enabled bucket, all versions of that object remain in the bucket; however, Amazon S3 inserts a delete marker. For more information, go to [Object Versioning](#).

The following scenarios describe the behavior of a Multi-Object Delete request when versioning is enabled for your bucket.

### Case 1 - Simple Delete

The following sample the Multi-Object Delete request specifies only one key.

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIee=
Content-Length: 79
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>SampleDocument.txt</Key>
  </Object>
</Delete>
```

Because versioning is enabled on the bucket, Amazon S3 does not delete the object. Instead, it adds a delete marker for this object. The response indicates that a delete marker was added (the `DeleteMarker` element in the response as a value of true) and the version number of the delete marker it added.

```
HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYxlrefdw3rEzmJh8z5KDtgzb+/FB7oiQaSci9Yaxd8o1YXc7d1111ab+
x-amz-request-id: 264A17BF16E9E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 276

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Deleted>
    <Key>SampleDocument.txt</Key>
    <DeleteMarker>true</DeleteMarker>
```

```
<DeleteMarkerVersionId>NeQt5xeFTfgPJD8B4CGWnkSLtluMr11s</DeleteMarkerVersionId>
</Deleted>
</DeleteResult>
```

## Case 2 - Versioned Delete

The following Multi-Object Delete attempts to delete a specific version of an object

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */*
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIxx=
Content-Length: 140
Connection: Keep-Alive

<Delete>
  <Object>
    <Key>SampleDocument.txt</Key>
    <VersionId>OYcLXagmS.WaD..oyH4KRguB95_YhLs7</VersionId>
  </Object>
</Delete>
```

In this case, Amazon S3 deletes the specific object version from the bucket and returns the following response. In the response, Amazon S3 returns the key and version ID of the object deleted.

```
HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYxlrefdw3rEzmJh8z5KDtgzb+/FB7oiQaSci9Yaxd8oLYXc7d1111xx+
x-amz-request-id: 264A17BF16E9E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 219

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Deleted>
    <Key>SampleDocument.txt</Key>
    <VersionId>OYcLXagmS.WaD..oyH4KRguB95_YhLs7</VersionId>
  </Deleted>
</DeleteResult>
```

## Case 3 - Versioned Delete of a Delete Marker

In the preceding example, the request refers to a delete marker (instead of an object), then Amazon S3 deletes the delete marker. The effect of this operation is to make your object reappear in your bucket. Amazon S3 returns a response that indicates the delete marker it deleted (`DeleteMarker` element with value true) and the version ID of the delete marker.

```
HTTP/1.1 200 OK
x-amz-id-2: IIPUZrtolxDEmWsKOae9J1Sz6yWfTye3HQ3T2iAe0ZE4XHa6NKvAJcPp51zzBr
x-amz-request-id: D6B284CEC9B05E4E
Date: Wed, 30 Nov 2011 03:43:25 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 331

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Deleted>
```

```
<Key>SampleDocument.txt</Key>
<VersionId>NeQt5xeFTfgPJD8B4CGWnkSLtluMr11s</VersionId>
<DeleteMarker>true</DeleteMarker>
<DeleteMarkerVersionId>NeQt5xeFTfgPJD8B4CGWnkSLtluMr11s</DeleteMarkerVersionId>
</Deleted>
</DeleteResult>
```

In general, when a Multi-Object Delete request results in Amazon S3 either adding a delete marker or removing a delete marker, the response returns the following elements.

### Example

```
<DeleteMarker>true</DeleteMarker>
<DeleteMarkerVersionId>NeQt5xeFTfgPJD8B4CGWnkSLtluMr11s</DeleteMarkerVersionId>
```

## Example 3: Malformed XML in the Request

This example shows how Amazon S3 responds to a request that includes a malformed XML document.

### Sample Request

The following requests sends a malformed XML document (missing the Delete end element).

```
POST /?delete HTTP/1.1
Host: bucketname.s3.amazonaws.com
Accept: */
x-amz-date: Wed, 30 Nov 2011 03:39:05 GMT
Content-MD5: p5/WA/oEr30qrEEl21PAqw==
Authorization: AWS AKIAIOSFODNN7EXAMPLE:W0qPYCLe6JwkZAD1ei6hp9XZIee=
Content-Length: 104
Connection: Keep-Alive

<Delete>
<Object>
<Key>404.txt</Key>
</Object>
<Object>
<Key>a.txt</Key>
</Object>
```

### Sample Response

The response returns the Error messages that describe the error.

```
HTTP/1.1 200 OK
x-amz-id-2: P3xqrhuhYxlrefdw3rEzmJh8z5KDtgzb+/FB7oiQaScI9Yaxd8oLYXc7d1111ab+
x-amz-request-id: 264A17BF16E9E80A
Date: Wed, 30 Nov 2011 03:39:32 GMT
Content-Type: application/xml
Server: AmazonS3
Content-Length: 207

<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>MalformedXML</Code>
<Message>The XML you provided was not well-formed or did not
validate against our published schema</Message>
<RequestId>264A17BF16E9E80A</RequestId>
<HostId>P3xqrhuhYxlrefdw3rEzmJh8z5KDtgzb+/FB7oiQaScI9Yaxd8oLYXc7d1111ab+</HostId>
```

| </Error>

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)
- [Upload Part \(p. 508\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Parts \(p. 502\)](#)

# DELETE Object

## Description

The `DELETE` operation removes the null version (if there is one) of an object and inserts a delete marker, which becomes the current version of the object. If there isn't a null version, Amazon S3 does not remove any objects.

## Versioning

To remove a specific version, you must be the bucket owner and you must use the `versionId` subresource. Using this subresource permanently deletes the version. If the object deleted is a delete marker, Amazon S3 sets the response header, `x-amz-delete-marker`, to `true`.

If the object you want to delete is in a bucket where the bucket versioning configuration is MFA Delete enabled, you must include the `x-amz-mfa` request header in the `DELETE` `versionId` request. Requests that include `x-amz-mfa` must use HTTPS.

For more information about MFA Delete, go to [Using MFA Delete](#). To see sample requests that use versioning, see [Sample Request \(p. 345\)](#).

You can delete objects by explicitly calling the `DELETE` Object API or configure its lifecycle (see [PUT Bucket lifecycle \(p. 265\)](#)) to enable Amazon S3 to remove them for you. If you want to block users or accounts from removing or deleting objects from your bucket you must deny them `s3:DeleteObject`, `s3:DeleteObjectVersion` and `s3:PutLifeCycleConfiguration` actions.

## Requests

### Syntax

```
DELETE /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Content-Length: length
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

Name	Description	Required
<code>x-amz-mfa</code>	<p>The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device.</p> <p>Type: String</p> <p>Default: None</p> <p>Condition: Required to permanently delete a versioned object if versioning is configured with MFA Delete enabled.</p>	Conditional

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

Header	Description
x-amz-delete-marker	<p>Specifies whether the versioned object that was permanently deleted was (true) or was not (false) a delete marker. In a simple DELETE, this header indicates whether (true) or not (false) a delete marker was created.</p> <p>Type: Boolean</p> <p>Valid Values: true   false</p> <p>Default: false</p>
x-amz-version-id	<p>Returns the version ID of the delete marker created as a result of the DELETE operation. If you delete a specific object version, the value returned by this header is the version ID of the object version deleted.</p> <p>Type: String</p> <p>Default: None</p>

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request deletes the object, my-second-image.jpg.

```
DELETE /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

### Sample Response

```
HTTP/1.1 204 NoContent
```

```
x-amz-id-2: LriYPLdmOdAiiI1fgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: OA49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Sample Request Deleting a Specified Version of an Object

The following request deletes the specified version of the object, my-third-image.jpg.

```
DELETE /my-third-image.jpg?versionId=UIORUnfndfiufdisoahr398493jfdkjFJjkndnqUifhnw89493jJFJ
HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 0
```

## Sample Response

```
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiiI1fgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: OA49CE4060975EAC
x-amz-version-id: UIORUnfndfiufdisoahr398493jfdkjFJjkndnqUifhnw89493jJFJ
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Sample Response if the Object Deleted is a Delete Marker

```
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiiI1fgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: OA49CE4060975EAC
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
x-amz-delete-marker: true
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Sample Request Deleting a Specified Version of an Object in an MFA-Enabled Bucket

The following request deletes the specified version of the object, my-third-image.jpg, which is stored in an MFA-enabled bucket.

```
DELETE /my-third-image.jpg?versionId=UIORUnfndfiuf HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-mfa: [SerialNumber] [AuthenticationCode]
Authorization: authorization string
Content-Type: text/plain
Content-Length: 0
```

## Sample Response

```
HTTPS/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: UIORUnfndfiuf
Date: Wed, 12 Oct 2009 17:50:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Object \(p. 412\)](#)
- [DELETE Object \(p. 343\)](#)

# DELETE Object tagging

## Description

This implementation of the `DELETE` operation uses the `tagging` subresource to remove the entire tag set from the specified object. For more information about managing object tags, see [Object Tagging](#) in the *Amazon Simple Storage Service Developer Guide*.

To use this operation, you must have permission to perform the `s3:DeleteObjectTagging` action.

To delete tags of a specific object version, add the `versionId` query parameter in the request. You will need permission for the `s3:DeleteObjectVersionTagging` action.

## Requests

### Syntax

```
DELETE ObjectKey?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Examples

### Sample Request

The following `DELETE` request deletes the tag set from the specified object.

```
DELETE exampleobject/?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```

```
Date: Wed, 25 Nov 2016 12:00:00 GMT
Authorization: signatureValue
```

## Sample Response

The following successful response shows Amazon S3 returning a 204 No Content response. The tag set for the object has been removed.

```
HTTP/1.1 204 No Content
Date: Wed, 25 Nov 2016 12:00:00 GMT
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Object tagging \(p. 454\)](#)
- [GET Object tagging \(p. 368\)](#)

# GET Object

## Description

This implementation of the `GET` operation retrieves objects from Amazon S3. To use `GET`, you must have `READ` access to the object. If you grant `READ` access to the anonymous user, you can return the object without using an authorization header.

An Amazon S3 bucket has no directory hierarchy such as you would find in a typical computer file system. You can, however, create a logical hierarchy by using object key names that imply a folder structure. For example, instead of naming an object `sample.jpg`, you can name it `photos/2006/February/sample.jpg`.

To get an object from such a logical hierarchy, specify the full key name for the object in the `GET` operation. For a virtual hosted-style request example, if you have the object `photos/2006/February/sample.jpg`, specify the resource as `/photos/2006/February/sample.jpg`. For a path-style request example, if you have the object `photos/2006/February/sample.jpg` in the bucket named `examplebucket`, specify the resource as `/examplebucket/photos/2006/February/sample.jpg`. For more information about request types, see [HTTP Host Header Bucket Specification](#) in the *Amazon Simple Storage Service Developer Guide*.

To distribute large files to many people, you can save bandwidth costs by using BitTorrent. For more information, see [Amazon S3 Torrent](#) in the *Amazon Simple Storage Service Developer Guide*. For more information about returning the ACL of an object, see [GET Object ACL \(p. 361\)](#).

If the object you are retrieving is a `GLACIER` storage class object, the object is archived in Glacier. You must first restore a copy using the [POST Object restore \(p. 397\)](#) API before you can retrieve the object. Otherwise, this operation returns an `InvalidObjectStateException` error. For information about archiving objects in Glacier, go to [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

If you encrypt an object by using server-side encryption with customer-provided encryption keys (SSE-C) when you store the object in Amazon S3, then when you `GET` the object, you must use the headers documented in the section [Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys \(p. 353\)](#). For more information about SSE-C, go to [Server-Side Encryption \(Using Customer-Provided Encryption Keys\)](#) in the *Amazon Simple Storage Service Developer Guide*.

Assuming you have permission to read object tags (permission for the `s3:GetObjectVersionTagging` action), the response also returns the `x-amz-tagging-count` header that provides the count of number of tags associated with the object. You can use the "GET Object tagging" API (see [GET Object tagging \(p. 368\)](#)) to retrieve the tag set associated with an object.

## Permissions

You need the `s3:GetObject` permission for this operation. For more information, go to [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*. If the object you request does not exist, the error Amazon S3 returns depends on whether you also have the `s3>ListBucket` permission.

- If you have the `s3>ListBucket` permission on the bucket, Amazon S3 will return an HTTP status code 404 ("no such key") error.
- if you don't have the `s3>ListBucket` permission, Amazon S3 will return an HTTP status code 403 ("access denied") error.

## Versioning

By default, the `GET` operation returns the current version of an object. To return a different version, use the `versionId` subresource.

### Note

If the current version of the object is a delete marker, Amazon S3 behaves as if the object was deleted and includes `x-amz-delete-marker: true` in the response.

For more information about versioning, see [PUT Bucket versioning \(p. 317\)](#) To see sample requests that use versioning, see [Sample Request Getting a Specified Version of an Object \(p. 358\)](#).

## Requests

### Syntax

```
GET /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Range:bytes=byte_range
```

### Request Parameters

There are times when you want to override certain response header values in a `GET` response. For example, you might override the `Content-Disposition` response header value in your `GET` request.

You can override values for a set of response headers using the query parameters listed in the following table. These response header values are sent only on a successful request, that is, when status code `200 OK` is returned. The set of headers you can override using these parameters is a subset of the headers that Amazon S3 accepts when you create an object. The response headers that you can override for the `GET` response are `Content-Type`, `Content-Language`, `Expires`, `Cache-Control`, `Content-Disposition`, and `Content-Encoding`. To override these header values in the `GET` response, you use the request parameters described in the following table.

### Note

You must sign the request, either using an `Authorization` header or a presigned URL, when using these parameters. They cannot be used with an unsigned (anonymous) request.

Parameter	Description	Required
<code>response-content-type</code>	Sets the <code>Content-Type</code> header of the response.  Type: String  Default: None	No
<code>response-content-language</code>	Sets the <code>Content-Language</code> header of the response.  Type: String  Default: None	No
<code>response-expires</code>	Sets the <code>Expires</code> header of the response.  Type: String	No

Parameter	Description	Required
	Default: None	
<code>response-cache-control</code>	Sets the Cache-Control header of the response.  Type: String  Default: None	No
<code>response-content-disposition</code>	Sets the Content-Disposition header of the response.  Type: String  Default: None	No
<code>response-content-encoding</code>	Sets the Content-Encoding header of the response.  Type: String  Default: None	No

## Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
<code>Range</code>	Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35</a> .  Type: String  Default: None  Constraints: None	No
<code>If-Modified-Since</code>	Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).  See Consideration 2 after the table.  Type: String  Default: None  Constraints: None	No
<code>If-Unmodified-Since</code>	Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).  See Consideration 1 after the table.  Type: String  Default: None	No

Name	Description	Required
	Constraints: None	
If-Match	<p>Return the object only if its entity tag (<code>ETag</code>) is the same as the one specified; otherwise, return a 412 (precondition failed).</p> <p>See Consideration 1 after the table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
If-None-Match	<p>Return the object only if its entity tag (<code>ETag</code>) is different from the one specified; otherwise, return a 304 (not modified).</p> <p>See Consideration 2 after the table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No

#### Note

Encryption request headers, like `x-amz-server-side-encryption`, should not be sent for GET requests if your object uses server-side encryption with AWS KMS-managed encryption keys (SSE-KMS) or server-side encryption with Amazon S3-managed encryption keys (SSE-S3). If your object does use these types of keys, you'll get an HTTP 400 BadRequest error.

Note the following additional considerations about the preceding request headers:

- **Consideration 1** – If both of the `If-Match` and `If-Unmodified-Since` headers are present in the request as follows:

`If-Match` condition evaluates to `true`, and;

`If-Unmodified-Since` condition evaluates to `false`;

then, S3 returns 200 `OK` and the data requested. For more information about conditional requests, see [RFC 7232](#).

- **Consideration 2** – If both of the `If-None-Match` and `If-Modified-Since` headers are present in the request as follows:

`If-None-Match` condition evaluates to `false`, and;

`If-Modified-Since` condition evaluates to `true`;

then, S3 returns 304 `Not Modified` response code. For more information about conditional requests, see [RFC 7232](#).

## Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys

When you retrieve an object from Amazon S3 that was encrypted by using server-side encryption with customer-provided encryption keys (SSE-C), you must use the following request headers. For more information about SSE-C, go to [Server-Side Encryption \(Using Customer-Provided Encryption Keys\)](#) in the *Amazon Simple Storage Service Developer Guide*.

Name	Description	Required
x-amz-server-side-encryption-customer-algorithm	<p>Specifies the algorithm to use to when decrypting the requested object.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Values: AES256</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
x-amz-server-side-encryption-customer-key	<p>Specifies the customer-provided base64-encoded encryption key to use to decrypt the requested object. This value is used to perform the decryption and then it is discarded; Amazon does not store the key. The key must be appropriate for use with the algorithm specified in the <code>x-amz-server-side-encryption-customer-algorithm</code> header.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
x-amz-server-side-encryption-customer-key-MD5	<p>Specifies the base64-encoded 128-bit MD5 digest of the customer-provided encryption key according to <a href="#">RFC 1321</a>. If this header is included in your request, Amazon S3 uses it for a message integrity check to ensure that the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</p>	No

## Request Elements

This implementation of the operation does not use request elements.

# Responses

## Response Headers

Header	Description
x-amz-delete-marker	<p>Specifies whether the object retrieved was (<code>true</code>) or was not (<code>false</code>) a delete marker. If <code>false</code>, this response header does not appear in the response.</p> <p>Type: Boolean</p> <p>Valid Values: <code>true</code>   <code>false</code></p> <p>Default: <code>false</code></p>
x-amz-expiration	<p>Amazon S3 returns this header if an <code>Expiration</code> action is configured for the object as part of the bucket's lifecycle configuration. The header value includes an "expiry-date" component and a URL-encoded "rule-id" component. Note that for versioning-enabled buckets, this header applies only to current versions; Amazon S3 does not provide a header to infer when a noncurrent version will be eligible for permanent deletion. For more information, see <a href="#">PUT Bucket lifecycle (p. 265)</a>.</p> <p>Type: String</p>
x-amz-meta-*	<p>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata.</p> <p>Type: String</p>
x-amz-replication-status	<p>Amazon S3 can return this header if your request involves a bucket that is either a source or destination in a cross-region replication.</p> <p>In cross-region replication you have a source bucket on which you configure replication and destination bucket where Amazon S3 stores object replicas. When you request an object (GET Object) or object metadata (HEAD Object) from these buckets, Amazon S3 will return the <code>x-amz-replication-status</code> header in the response as follow:</p> <ul style="list-style-type: none"> <li>If requesting object from the source bucket — Amazon S3 will return the <code>x-amz-replication-status</code> header if object in your request is eligible for replication.</li> </ul> <p>For example, suppose in your replication configuration you specify object prefix "TaxDocs" requesting Amazon S3 to replicate objects with key prefix "TaxDocs". Then any objects you upload with this key name prefix, for example "TaxDocs/document1.pdf", is eligible for replication. For any object request with this key name prefix Amazon S3 will return the <code>x-amz-replication-status</code> header with value PENDING, COMPLETED or FAILED indicating object replication status.</p> <ul style="list-style-type: none"> <li>If requesting object from the destination bucket — Amazon S3 will return the <code>x-amz-replication-status</code> header with value REPLICA if object in your request is a replica that Amazon S3 created.</li> </ul> <p>For more information, go to <a href="#">Cross-Region Replication</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>

Header	Description
	<p>Valid Values: PENDING, COMPLETED, FAILED, REPLICA</p> <p>Type: String</p>
<b>x-amz-server-side-encryption</b>	If the object is stored using server-side encryption either with an AWS KMS or an Amazon S3-managed encryption key, the response includes this header with the value of the encryption algorithm used.
	<p>Type: String</p>
<b>x-amz-server-side-encryption-aws-kms-key-id</b>	If the <b>x-amz-server-side-encryption</b> is present and has the value of <code>aws:kms</code> , this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.
	<p>Type: String</p>
<b>x-amz-server-side-encryption-customer-algorithm</b>	If server-side encryption with customer-provided encryption keys decryption was requested, the response will include this header confirming the decryption algorithm used.
	<p>Type: String</p> <p>Valid Values: AES256</p>
<b>x-amz-server-side-encryption-customer-key-MD5</b>	If server-side encryption with customer-provided encryption keys decryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key.
	<p>Type: String</p>
<b>x-amz-storage-class</b>	<p>Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.</p> <p>For more information, go to <a href="#">Storage Classes</a> in <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p>
<b>x-amz-restore</b>	<p>Provides information about the object restoration operation and expiration time of the restored object copy.</p> <p>For more information about archiving objects and restoring them, go to <a href="#">Transitioning Objects: General Considerations</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p>
<b>x-amz-tagging-count</b>	<p>Returns the count of the tags associated with the object. This header is returned only if the count is greater than zero.</p> <p>Type: String</p> <p>Default: None</p>

Header	Description
<code>x-amz-version-id</code>	Returns the version ID of the retrieved object if it has a unique version ID.  Type: String  Default: None
<code>x-amz-website-redirect-location</code>	When a bucket is configured as a website, you can set this metadata on the object so the website endpoint will evaluate the request for the object as a 301 redirect to another object in the same bucket or an external URL.  Type: String  Default: None
<code>x-amz-object-lock-mode</code>	The Object Lock mode, if any, that's in effect for this object. This header is only returned if the requester has the <code>s3:GetObjectRetention</code> permission. For more information about S3 Object Lock, see <a href="#">Object Lock</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  Type: String  Valid values: GOVERNANCE   COMPLIANCE
<code>x-amz-object-lock-retain-until-date</code>	The date and time when the Object Lock retention period expires. This header is only returned if the requester has the <code>s3:GetObjectRetention</code> permission.  Type: Timestamp  Format: <code>2020-01-05T00:00:00.000Z</code>
<code>x-amz-object-lock-legal-hold</code>	Specifies whether a legal hold is in effect for this object. This header is only returned if the requester has the <code>s3:GetObjectLegalHold</code> permission. This header is not returned if the specified version of this object has never had a legal hold applied. For more information about legal holds, see <a href="#">Object Lock</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  Type: String  Valid values: ON   OFF

## Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the object, `my-image.jpg`.

```
GET /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Mon, 3 Oct 2016 22:32:00 GMT
Authorization: authorization string
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234

[434234 bytes of object data]
```

If the object had tags associated with it, S3 returns the `x-amz-tagging-count` header with tag count.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
x-amz-tagging-count: 2

[434234 bytes of object data]
```

If the object had expiration set using lifecycle configuration, you get the following response with the `x-amz-expiration` header.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-expiration: expiry-date="Fri, 23 Dec 2012 00:00:00 GMT", rule-id="picture-deletion-rule"
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain

[434234 bytes of object data]
```

## Sample Response if an Object Is Archived in Glacier

An object archived in Glacier must first be restored before you can access it. If you attempt to access an Glacier object without restoring it, Amazon S3 returns the following error.

```
HTTP/1.1 403 Forbidden
x-amz-request-id: CD4BD8A1310A11B3
x-amz-id-2: m9RDbQU0+RRBTjOUN1ChQ1eqMUnr9dv8b+KP6I2gHfRJZSTSrMCoRP8RtPRzx9mb
Content-Type: application/xml
Date: Mon, 12 Nov 2012 23:53:21 GMT
Server: AmazonS3
Content-Length: 231
```

```
<Error>
<Code>InvalidObjectState</Code>
<Message>The operation is not valid for the object's storage class</Message>
<RequestId>9FFF118E15B86F</RequestId>
<HostId>WVQ5kzhIT+oiUfDCOioYv8W4Tk9eNcxWi/MK+hTS/av34Xy4rBU3zsavf0aaaaa</HostId>
</Error>
```

## Sample Response if the Latest Object Is a Delete Marker

```
HTTP/1.1 404 Not Found
x-amz-request-id: 318BC8BC148832E5
x-amz-id-2: eftixk72aD6Ap51Tnqzj7UDNEHGran
x-amz-version-id: 3GL4kqtJlcpXroDTDm3vjVBH40Nr8X8g
x-amz-delete-marker: true
Date: Wed, 28 Oct 2009 22:32:00 GMT
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

Notice that the delete marker returns a 404 Not Found error.

## Sample Request Getting a Specified Version of an Object

The following request returns the specified version of an object.

```
GET /myObject?versionId=3/L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

## Sample Response to a Versioned Object GET Request

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap54OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3QBpUMLUo
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
[434234 bytes of object data]
```

## Sample Request with Parameters Altering Response Header Values

The following request specifies all the query string parameters in a GET request overriding the response header values.

```
GET /Junk3.txt?response-cache-control=No-cache&response-content-disposition=attachment%3B
%20filename%3Dtesting.txt&response-content-encoding=x-gzip&response-content-language=mi%2C
%20en&response-expires=Thu%2C%2001%20Dec%201994%2016:00:00%20GMT HTTP/1.1
x-amz-date: Sun, 19 Dec 2010 01:53:44 GMT
Accept: */*
```

```
Authorization: AWS AKIAIOSFODNN7EXAMPLE:aaStE6nKnw8ihhiIdReoXYlMamW=
```

## Sample Response with Overridden Response Header Values

In the following sample response note, the header values are set to the values specified in the `true` request.

```
HTTP/1.1 200 OK
x-amz-id-2: SIidWAK3hK+I13/Qgiu1ZKEuegzLAAspwsgwnwygb9GgFseeFHL5CII8NXSrFWW2
x-amz-request-id: 881B1CBD9DF17WA1
Date: Sun, 19 Dec 2010 01:54:01 GMT
x-amz-meta-param1: value 1
x-amz-meta-param2: value 2
Cache-Control: No-cache
Content-Language: mi, en
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Disposition: attachment; filename=testing.txt
Content-Encoding: x-gzip
Last-Modified: Fri, 17 Dec 2010 18:10:41 GMT
ETag: "0332bee1a7bf845f176c5c0d1ae7cf07"
Accept-Ranges: bytes
Content-Type: text/plain
Content-Length: 22
Server: AmazonS3

[object data not shown]
```

## Sample Request with a Range Header

The following request specifies the HTTP Range header to retrieve the first 10 bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sect14.html>.

```
GET /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
x-amz-date: Fri, 28 Jan 2011 21:32:02 GMT
Range: bytes=0-9
Authorization: AWS AKIAIOSFODNN7EXAMPLE:Yxg83MzaEgh3OZ3l0rLo5RTX11o=
Sample Response with Specified Range of the Object Bytes
```

### Note

Amazon S3 doesn't support retrieving multiple ranges of data per GET request.

## Sample Response

In the following sample response, note that the header values are set to the values specified in the `true` request.

```
HTTP/1.1 206 Partial Content
x-amz-id-2: MzRIS0wyjmnuCzjI1WC0615TTAzm7/JypPGXLh0OVFGcJaaO3KW/hRAqKOPIEEp
x-amz-request-id: 47622117804B3E11
Date: Fri, 28 Jan 2011 21:32:09 GMT
x-amz-meta-title: the title
Last-Modified: Fri, 28 Jan 2011 20:10:32 GMT
ETag: "b2419b1e3fd45d596ee22bdf62aaaa2f"
Accept-Ranges: bytes
Content-Range: bytes 0-9/443
Content-Type: text/plain
Content-Length: 10
```

```
Server: AmazonS3  
[10 bytes of object data]
```

## Sample: Get an Object Stored Using Server-Side Encryption with Customer-Provided Encryption Keys

If an object is stored in Amazon S3 using server-side encryption with customer-provided encryption keys, Amazon S3 needs encryption information so that it can decrypt the object before sending it to you in response to a GET request. You provide the encryption information in your GET request using the relevant headers (see [Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys \(p. 353\)](#)), as shown in the following example request.

```
GET /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com

Accept: /*
Authorization:authorization string
Date: Wed, 28 May 2014 19:24:44 +0000
x-amz-server-side-encryption-customer-key:g01CfA3Dv40jZz5SQJ1ZukLRFqtI5WorC/8SEKEXAMPLE
x-amz-server-side-encryption-customer-key-MD5:ZjQrne1X/iTcskbY2m3example
x-amz-server-side-encryption-customer-algorithm:AES256
```

The following sample response shows some of the response headers Amazon S3 returns. Note that it includes the encryption information in the response.

```
HTTP/1.1 200 OK
x-amz-id-2: ka5jRm8X3N12ZiY29Z989zg2tNSJPMcK+to7jNjxImXBbyChqc6tLAv+sau7Vjzh
x-amz-request-id: 195157E3E073D3F9
Date: Wed, 28 May 2014 19:24:45 GMT
Last-Modified: Wed, 28 May 2014 19:21:01 GMT
ETag: "c12022c9a3c6d3a28d29d90933a2b096"
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2m3example
```

## Related Resources

- [GET Service \(p. 65\)](#)
- [GET Object ACL \(p. 361\)](#)

# GET Object ACL

## Description

This implementation of the `GET` operation uses the `acl` subresource to return the access control list (ACL) of an object. To use this operation, you must have `READ_ACP` access to the object.

## Versioning

By default, `GET` returns ACL information about the current version of an object. To return ACL information about a different version, use the `versionId` subresource.

To see sample requests that use Versioning, see [Sample Request Getting the ACL of the Specific Version of an Object \(p. 363\)](#).

## Requests

### Syntax

```
GET /ObjectName?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Range:bytes=byte_range
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
<code>AccessControlList</code>	Container for Grant, Grantee, and Permission.

Name	Description
	Type: Container  Ancestors: AccessControlPolicy
AccessControlPolicy	Contains the elements that set the ACL permissions for an object per Grantee.  Type: Container  Ancestors: None
DisplayName	Screen name of the bucket owner.  <b>Important</b> This value is only included in the response in the US East (N. Virginia), US West (N. California), US West (Oregon), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), EU (Ireland), and South America (São Paulo) regions. For a list of all the Amazon S3 supported regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .  Type: String  Ancestors: AccessControlPolicy.Owner
Grant	Container for the grantee and his or her permissions.  Type: Container  Ancestors: AccessControlPolicy.AccessControlList
Grantee	The subject whose permissions are being set.  Type: String  Ancestors: AccessControlPolicy.AccessControlList.Grant
ID	ID of the bucket owner, or the ID of the grantee.  Type: String  Ancestors: AccessControlPolicy.Owner or AccessControlPolicy.AccessControlList.Grant
Owner	Container for the bucket owner's display name and ID.  Type: Container  Ancestors: AccessControlPolicy
Permission	Specifies the permission (FULL_CONTROL, WRITE, READ_ACP) given to the grantee.  Type: String  Ancestors: AccessControlPolicy.AccessControlList.Grant

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns information, including the ACL, of the object, my-image.jpg.

```
GET /my-image.jpg?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 4HL4kqtJlcpXroDTDmJ+rmSpXd3d1brHY+MTRCx3vVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
        <DisplayName>mtd@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

### Sample Request Getting the ACL of the Specific Version of an Object

The following request returns information, including the ACL, of the specified version of the object, my-image.jpg.

```
GET /my-image.jpg?versionId=3/L4kqtJlcpXroDVBH40Nr8X8gdRQBpUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

## Sample Response Showing the ACL of the Specific Version

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCx3vjbVH40Nr8X8gdRQBpUMLUo
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mdtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
        <DisplayName>mdtd@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

## Related Resources

- [GET Object \(p. 349\)](#)
- [PUT Object \(p. 412\)](#)
- [DELETE Object \(p. 343\)](#)

# GET Object legal hold

Service: Amazon Simple Storage Service

Gets an object's current Legal Hold status.

## Request Syntax

```
GET /<object-key>?legal-hold&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests \(AWS Signature Version 4\))
```

## URI Request Parameters

### versionId

The version ID for the object version whose retention settings you want to retrieve.

## Request Body

The request does not have a request body.

## Response Syntax

```
HTTP/1.1 200
<?xml version="1.0" encoding="UTF-8"?>
<LegalHold>
    <Status>string</Status>
</LegalHold>
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

### LegalHold ([p. 365](#))

Root level tag for the LegalHold parameters.

### Status ([p. 365](#))

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

## Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

# GET Object retention

Service: Amazon Simple Storage Service

Retrieves an object's retention settings.

## Request Syntax

```
GET /<object-key>?retention&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests \(AWS Signature Version 4\))
```

## URI Request Parameters

### versionId

The version ID for the object version whose retention settings you want to retrieve.

## Request Body

The request does not have a request body.

## Response Syntax

```
<Retention>
  <Mode>string</Mode>
  <RetainUntilDate>timestamp</RetainUntilDate>
</Retention>
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in XML format by the service.

### Retention (p. 366)

Root level tag for the Retention parameters.

#### Mode (p. 366)

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

#### RetainUntilDate (p. 366)

Type: Timestamp

Format: **2020-01-05T00:00:00.000Z**

## Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

# GET Object tagging

## Description

This implementation of the GET operation returns the tags associated with an object. You send the GET request against the tagging subresource associated with the object.

To use this operation, you must have permission to perform the s3:GetObjectTagging action. By default, the GET operation returns information about current version of an object. For a versioned bucket, you can have multiple versions of an object in your bucket. To retrieve tags of any other version, use the `versionId` query parameter. You also need permission for the s3:GetObjectVersionTagging action.

By default, the bucket owner has this permission and can grant this permission to others.

For information about the Amazon S3 object tagging feature, see [Object Tagging](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /ObjectName?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
Tagging	Container for the TagSet element.

Name	Description
	Type: Container  Ancestors: None
TagSet	Contains the tag set.  Type: Container  Ancestors: Tagging
Tag	Contains the tag information.  Type: Container  Ancestors: TagSet
Key	Name of the tag  Type: String  Ancestors: Tag
Value	Value of the tag  Type: String  Ancestors: Tag

## Examples

### Sample Request

The following request returns the tag set of the specified object.

```
GET /example-object?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Thu, 22 Sep 2016 21:33:08 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 22 Sep 2016 21:33:08 GMT
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>
<Tagging xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
```

| </Tagging>

## Related Resources

- [PUT Object tagging \(p. 454\)](#)

# GET Object torrent

## Description

This implementation of the `GET` operation uses the `torrent` subresource to return torrent files from a bucket. BitTorrent can save you bandwidth when you're distributing large files. For more information about BitTorrent, see [Amazon S3 Torrent](#).

### Note

You can get torrent only for objects that are less than 5 GB in size and that are not encrypted using server-side encryption with customer-provided encryption key.

To use `GET`, you must have `READ` access to the object.

## Requests

### Syntax

```
GET /ObjectName?torrent HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Getting Torrent Files in a Bucket

This example retrieves the Torrent file for the "Nelson" object in the "quotes" bucket.

```
GET /quotes/Nelson?torrent HTTP/1.0
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-request-id: 7CD745EBB7AB5ED9
Date: Wed, 25 Nov 2009 12:00:00 GMT
Content-Disposition: attachment; filename=Nelson.torrent;
Content-Type: application/x-bittorrent
Content-Length: 537
Server: AmazonS3

<body: a Bencoded dictionary as defined by the BitTorrent specification>
```

### Related Resources

- [GET Object \(p. 349\)](#)

# HEAD Object

## Description

The `HEAD` operation retrieves metadata from an object without returning the object itself. This operation is useful if you are interested only in an object's metadata. To use `HEAD`, you must have `READ` access to the object.

A `HEAD` request has the same options as a `GET` operation on an object. The response is identical to the `GET` response except that there is no response body.

If you encrypt an object by using server-side encryption with customer-provided encryption keys (SSE-C) when you store the object in Amazon S3, then when you retrieve the metadata from the object, you must use the headers documented in the section [Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys \(p. 375\)](#). For more information about SSE-C, go to [Server-Side Encryption \(Using Customer-Provided Encryption Keys\)](#) in the *Amazon Simple Storage Service Developer Guide*.

## Permissions

You need the `s3:GetObject` permission for this operation. For more information, go to [Specifying Permissions in a Policy](#) in the *Amazon Simple Storage Service Developer Guide*. If the object you request does not exist, the error Amazon S3 returns depends on whether you also have the `s3:ListBucket` permission.

- If you have the `s3:ListBucket` permission on the bucket, Amazon S3 will return a HTTP status code 404 ("no such key") error.
- If you don't have the `s3:ListBucket` permission, Amazon S3 will return a HTTP status code 403 ("access denied") error.

## Versioning

By default, the `HEAD` operation retrieves metadata from the current version of an object. If the current version is a delete marker, Amazon S3 behaves as if the object was deleted. To retrieve metadata from a different version, use the `versionId` subresource. For more information, see [Versions](#) in the *Amazon Simple Storage Service Developer Guide*.

To see sample requests that use versioning, see [Sample Request Getting Metadata from a Specified Version of an Object \(p. 380\)](#).

## Requests

### Syntax

```
HEAD /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Date: date
```

### Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
Range	<p>Downloads the specified range bytes of an object. For more information about the HTTP Range header, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
If-Modified-Since	<p>Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).</p> <p>See Consideration 2 after the table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
If-Unmodified-Since	<p>Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).</p> <p>See Consideration 1 after the table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
If-Match	<p>Return the object only if its entity tag (<code>ETag</code>) is the same as the one specified; otherwise, return a 412 (precondition failed).</p> <p>See Consideration 1 after the table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
If-None-Match	<p>Return the object only if its entity tag (<code>ETag</code>) is different from the one specified; otherwise, return a 304 (not modified).</p> <p>See Consideration 2 after the table.</p> <p>Type: String</p> <p>Default: None</p>	No

Name	Description	Required
	Constraints: None	

**Note**

Encryption request headers, like `x-amz-server-side-encryption`, should not be sent for `GET` requests if your object uses server-side encryption with AWS KMS-managed encryption keys (SSE-KMS) or server-side encryption with Amazon S3-managed encryption keys (SSE-S3). If your object does use these types of keys, you'll get an HTTP 400 BadRequest error.

Note the following additional considerations about the preceding request headers:

- **Consideration 1** – If both of the `If-Match` and `If-Unmodified-Since` headers are present in the request as follows:

`If-Match` condition evaluates to `true`, and;

`If-Unmodified-Since` condition evaluates to `false`;

then, Amazon S3 returns 200 `OK` and the data requested. For more information about conditional requests, see [RFC 7232](#).

- **Consideration 2** – If both of the `If-None-Match` and `If-Modified-Since` headers are present in the request as follows:

`If-None-Match` condition evaluates to `false`, and;

`If-Modified-Since` condition evaluates to `true`;

then, Amazon S3 returns the 304 `Not Modified` response code. For more information about conditional requests, see [RFC 7232](#).

## Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys

When you retrieve metadata from an object stored in Amazon S3 that was encrypted by using server-side encryption with customer-provided encryption keys (SSE-C), you must use the following request headers. For more information about SSE-C, go to [Server-Side Encryption \(Using Customer-Provided Encryption Keys\)](#) in the *Amazon Simple Storage Service Developer Guide*.

Name	Description	Required
<code>x-amz-server-side-encryption-customer-algorithm</code>	<p>Specifies the algorithm to use to when decrypting the requested object.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Values: <code>AES256</code></p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes

Name	Description	Required
x-amz-server-side-encryption-customer-key	<p>Specifies the customer-provided base64-encoded encryption key to use to decrypt the requested object. This value is used to perform the decryption and then it is discarded; Amazon does not store the key. The key must be appropriate for use with the algorithm specified in the <code>x-amz-server-side-encryption-customer-algorithm</code> header.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
x-amz-server-side-encryption-customer-key-MD5	<p>Specifies the base64-encoded 128-bit MD5 digest of the customer-provided encryption key according to <a href="#">RFC 1321</a>. If this header is included in your request, Amazon S3 uses it for a message integrity check to ensure that the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</p>	No

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
x-amz-expiration	<p>Amazon S3 returns this header if an <code>Expiration</code> action is configured for the object as part of the bucket's lifecycle configuration. The header value includes an "expiry-date" component and a URL-encoded "rule-id" component. Note that for versioning-enabled buckets, this header applies only to current versions; Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see <a href="#">PUT Bucket lifecycle (p. 265)</a>.</p> <p>Type: String</p>

Name	Description
x-amz-meta-*	<p>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata.</p> <p>Type: String</p>
x-amz-missing-meta	<p>This header is set to the number of metadata entries that were not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, with SOAP, you can create metadata with values that are not valid HTTP headers.</p> <p>Type: String</p>
x-amz-replication-status	<p>Amazon S3 can return this header if your request involves a bucket that is either a source or destination in a cross-region replication.</p> <p>In cross-region replication, you have a source bucket on which you configure replication and destination bucket where Amazon S3 stores object replicas. When you request an object (GET Object) or object metadata (HEAD Object) from these buckets, Amazon S3 returns the x-amz-replication-status header in the response as follows:</p> <ul style="list-style-type: none"> <li>• If requesting object from the source bucket — Amazon S3 returns the x-amz-replication-status header if object in your request is eligible for replication.</li> </ul> <p>For example, suppose that in your replication configuration you specify object prefix "TaxDocs" requesting Amazon S3 to replicate objects with key prefix "TaxDocs". Then any objects you upload with this key name prefix, for example "TaxDocs/document1.pdf", is eligible for replication. For any object request with this key name prefix, Amazon S3 returns the x-amz-replication-status header with value PENDING, COMPLETED, or FAILED indicating object replication status.</p> <ul style="list-style-type: none"> <li>• If requesting object from the destination bucket — Amazon S3 returns the x-amz-replication-status header with value REPLICA if object in your request is a replica that Amazon S3 created.</li> </ul> <p>For more information, see <a href="#">Cross-Region Replication</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Valid Values: PENDING, COMPLETED, FAILED, REPLICA</p> <p>Type: String</p>

Name	Description
<code>x-amz-restore</code>	<p>If the object is an archived object (an object whose storage class is <code>GLACIER</code>), the response includes this header if either the archive restoration is in progress (see <a href="#">POST Object restore (p. 397)</a>) or an archive copy is already restored.</p> <p>If an archive copy is already restored, the header value indicates when Amazon S3 is scheduled to delete the object copy. For example,</p> <pre><code>x-amz-restore: ongoing-request="false", expiry-date="Fri, 23 Dec 2012 00:00:00 GMT"</code></pre> <p>If the object restoration is in progress, the header returns the value <code>ongoing-request="true"</code>.</p> <p>For more information about archiving objects, see <a href="#">Transitioning Objects: General Considerations</a> in the <i>Amazon Simple Storage Service Developer Guide</i></p> <p>Type: String</p> <p>Default: None</p>
<code>x-amz-server-side-encryption</code>	<p>If the object is stored using server-side encryption either with an AWS KMS or an Amazon S3-managed encryption key, the response includes this header with the value of the encryption algorithm used.</p> <p>Type: String</p>
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	<p>If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code>, this header specifies the ID of the AWS KMS master encryption key that was used for the object.</p> <p>Type: String</p>
<code>x-amz-server-side-encryption-customer-algorithm</code>	<p>If server-side encryption with customer-provided encryption keys(SSE-C) decryption was requested, the response includes this header confirming the decryption algorithm used.</p> <p>Type: String</p> <p>Valid Values: <code>AES256</code></p>
<code>x-amz-server-side-encryption-customer-key-MD5</code>	<p>If SSE-C decryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key.</p> <p>Type: String</p>
<code>x-amz-storage-class</code>	<p>Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.</p> <p>For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p>

Name	Description
x-amz-version-id	The version ID of the object returned.  Type: String
x-amz-object-lock-mode	The Object Lock mode, if any, that's in effect for this object. This header is only returned if the requester has the s3:GetObjectRetention permission. For more information about S3 Object Lock, see <a href="#">Object Lock</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  Type: String  Valid values: GOVERNANCE   COMPLIANCE
x-amz-object-lock-retain-until-date	The date and time when the Object Lock retention period expires. This header is only returned if the requester has the s3:GetObjectRetention permission.  Type: Timestamp  Format: <b>2020-01-05T00:00:00.000Z</b>
x-amz-object-lock-legal-hold	Specifies whether a legal hold is in effect for this object. This header is only returned if the requester has the s3:GetObjectLegalHold permission. This header is not returned if the specified version of this object has never had a legal hold applied. For more information about legal holds, see <a href="#">Object Lock</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  Type: String  Valid values: ON   OFF

## Response Elements

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request returns the metadata of an object.

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed4OpIszzj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXrof3vjbVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

If the object is scheduled to expire according to a lifecycle configuration set on the bucket, the response returns the `x-amz-expiration` tag with information about when Amazon S3 will delete the object. For more information, see [Transitioning Objects: General Considerations](#) in the *Amazon Simple Storage Service Developer Guide*.

```
HTTP/1.1 200 OK
x-amz-id-2: azQRZtQJ2m1P8R+TIsG9h0VuC/DmiSJmjXUMq7snk+LKSJeurtmfzSlGhR46GzSJ
x-amz-request-id: 0EFF61CCE3F24A26
Date: Mon, 17 Dec 2012 02:26:39 GMT
Last-Modified: Mon, 17 Dec 2012 02:14:10 GMT
x-amz-expiration: expiry-date="Fri, 21 Dec 2012 00:00:00 GMT", rule-id="Rule for
testfile.txt"
ETag: "54b0c58c7ce9f2a8b551351102ee0938"
Accept-Ranges: bytes
Content-Type: text/plain
Content-Length: 14
Server: AmazonS3
```

## Sample Request Getting Metadata from a Specified Version of an Object

The following request returns the metadata of the specified version of an object.

```
HEAD /my-image.jpg?versionId=3HL4kqCx3vjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0WpaBX5sCYVf1bNRuU=
```

## Sample Response to a Versioned HEAD Request

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8epIszzj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXrof3vjbVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

## Sample Request for an Glacier Object

For an archived object, the `x-amz-restore` header provides the date when the restored copy expires, as shown in the following response. Even if the object is stored in Glacier, all object metadata is still available.

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: 13 Nov 2012 00:28:38 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

## Sample Response - Glacier Object

If the object is already restored, the `x-amz-restore` header provides the date when the restored copy will expire, as shown in the following response.

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3Izs1NnwBZeujM19iI8UbxBmbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Tue, 13 Nov 2012 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
x-amz-restore: ongoing-request="false", expiry-date="Wed, 07 Nov 2012 00:00:00 GMT"
ETag: "1acccb31fcf202eba0c0f41fa2f09b4d7"
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

If the restoration is in progress, then the `x-amz-restore` header returns a message accordingly.

```
HTTP/1.1 200 OK
x-amz-id-2: b+V2mDiMHTdy1myoUBpctvmJl95H9U/OSUm/jRtHxjh0+pCk5SvByL4xu2TDv4GM
x-amz-request-id: E2E7B6AEE4E9BD2B
Date: Tue, 13 Nov 2012 00:43:32 GMT
Last-Modified: Sat, 20 Oct 2012 21:28:27 GMT
x-amz-restore: ongoing-request="true"
ETag: "1acccb31fcf202eba0c0f41fa2f09b4d7"
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

## Related Resources

- [GET Object \(p. 349\)](#)

# OPTIONS object

## Description

A browser can send this preflight request to Amazon S3 to determine if it can send an actual request with the specific origin, HTTP method, and headers.

Amazon S3 supports cross-origin resource sharing (CORS) by enabling you to add a `cors` subresource on a bucket. When a browser sends this preflight request, Amazon S3 responds by evaluating the rules that are defined in the `cors` configuration.

If `cors` is not enabled on the bucket, then Amazon S3 returns a 403 Forbidden response.

For more information about CORS, go to [Enabling Cross-Origin Resource Sharing](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
OPTIONS /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Origin: Origin
Access-Control-Request-Method: HTTPMethod
Access-Control-Request-Headers: RequestHeader
```

### Request Parameters

This operation does not introduce any specific request parameters, but it may contain any request parameters that are required by the actual request.

### Request Headers

Name	Description	Required
Origin	Identifies the origin of the cross-origin request to Amazon S3. For example, http://www.example.com.  Type: String  Default: None	Yes
Access-Control-Request-Method	Identifies what HTTP method will be used in the actual request.  Type: String  Default: None	Yes
Access-Control-Request-Headers	A comma-delimited list of HTTP headers that will be sent in the actual request.  For example, to put an object with server-side encryption, this preflight request will determine if it can include the <code>x-amz-server-side-encryption</code> header with the request.	No

Name	Description	Required
	Type: String  Default: None	

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

Header	Description
Access-Control-Allow-Origin	The origin you sent in your request. If the origin in your request is not allowed, Amazon S3 will not include this header in the response.  Type: String
Access-Control-Max-Age	How long, in seconds, the results of the preflight request can be cached.  Type: String
Access-Control-Allow-Methods	The HTTP method that was sent in the original request. If the method in the request is not allowed, Amazon S3 will not include this header in the response.  Type: String
Access-Control-Allow-Headers	A comma-delimited list of HTTP headers that the browser can send in the actual request. If any of the requested headers is not allowed, Amazon S3 will not include that header in the response, nor will the response contain any of the headers with the Access-Control prefix.  Type: String
Access-Control-Expose-Headers	A comma-delimited list of HTTP headers. This header provides the JavaScript client with access to these headers in the response to the actual request.  Type: String

## Response Elements

This implementation of the operation does not return response elements.

## Examples

### Example : Send a preflight OPTIONS request to a cors enabled bucket

A browser can send this preflight request to Amazon S3 to determine if it can send the actual PUT request from `http://www.example.com` origin to the Amazon S3 bucket named `examplebucket`.

#### Sample Request

```
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Origin: http://www.example.com
Access-Control-Request-Method: PUT
```

#### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: 6SvaESv3VULYPLik5LLl7lSPPtSnBvDdGmnk1X1HfUl7uS2m1DF6td6KWKNjYMXZ
x-amz-request-id: BDC4B83DF5096BBE
Date: Wed, 21 Aug 2012 23:09:55 GMT
Etag: "1f1a1af1f111111111c11aed1da1"
Access-Control-Allow-Origin: http://www.example.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-amz-request-id
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [GET Bucket cors \(p. 131\)](#)
- [DELETE Bucket cors \(p. 82\)](#)
- [PUT Bucket cors \(p. 248\)](#)

# POST Object

## Description

The `POST` operation adds an object to a specified bucket using HTML forms. `POST` is an alternate form of `PUT` that enables browser-based uploads as a way of putting objects in buckets. Parameters that are passed to `PUT` via HTTP Headers are instead passed as form fields to `POST` in the multipart/form-data encoded message body. You must have `WRITE` access on a bucket to add an object to it. Amazon S3 never stores partial objects: if you receive a successful response, you can be confident the entire object was stored.

Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object simultaneously, all but the last object written is overwritten.

To ensure that data is not corrupted traversing the network, use the `Content-MD5` form field. When you use this form field, Amazon S3 checks the object against the provided MD5 value. If they do not match, Amazon S3 returns an error. Additionally, you can calculate the MD5 value while posting an object to Amazon S3 and compare the returned `ETag` to the calculated MD5 value. The `ETag` only reflects changes to the contents of an object, not its metadata.

### Note

To configure your application to send the Request Headers before sending the request body, use the 100-continue HTTP status code. For `POST` operations, this helps you avoid sending the message body if the message is rejected based on the headers (for example, authentication failure or redirect). For more information on the 100-continue HTTP status code, go to Section 8.2.3 of <http://www.ietf.org/rfc/rfc2616.txt>.

You can optionally request server-side encryption where Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it for you when you access it. You have the option of providing your own encryption key or you can use the AWS-managed encryption keys. For more information, go to [Using Server-Side Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

## Versioning

If you enable versioning for a bucket, `POST` automatically generates a unique version ID for the object being added. Amazon S3 returns this ID in the response using the `x-amz-version-id` response header.

If you suspend versioning for a bucket, Amazon S3 always uses `null` as the version ID of the object stored in a bucket.

For more information about returning the versioning state of a bucket, see [GET Bucket \(Versioning Status\) \(p. 199\)](#).

Amazon S3 is a distributed system. If you enable versioning for a bucket and Amazon S3 receives multiple write requests for the same object simultaneously, all of the objects are stored.

To see sample requests that use versioning, see [Sample Request \(p. 395\)](#).

## Requests

### Syntax

```
POST / HTTP/1.1
Host: destinationBucket.s3.amazonaws.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"

acl
--9431149156168
Content-Disposition: form-data; name="tagging"

<Tagging><TagSet><Tag><Key>Tag Name</Key><Value>Tag Value</Value></Tag></TagSet></Tagging>
--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"

uuid
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

metadata
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"

access-key-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.jpg"
Content-Type: image/jpeg

file_content
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to Amazon S3
--9431149156168--
```

## Request Parameters

This implementation of the operation does not use request parameters.

## Form Fields

This operation can use the following form fields.

Name	Description	Required
AWSAccessKeyId	<p>The AWS access key ID of the owner of the bucket who grants an Anonymous user access for a request that satisfies the set of constraints in the policy.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Required if a policy document is included with the request.</p>	Conditional
acl	<p>Specifies an Amazon S3 access control list. If an invalid access control list is specified, an error is generated. For more information on ACLs, go to <a href="#">Access Control List (ACL) Overview</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: private</p> <p>Valid Values: <code>private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</code></p>	No
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	<p>REST-specific headers. For more information, see <a href="#">PUT Object (p. 412)</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
file	<p>File or text content.</p> <p>The file or text content must be the last field in the form.</p> <p>You cannot upload more than one file at a time.</p> <p>Type: File or text content</p> <p>Default: None</p>	Yes
key	<p>The name of the uploaded key.</p> <p>To use the file name provided by the user, use the <code> \${filename}</code> variable. For example, if the user Betty uploads the file <code>lolcatz.jpg</code> and you specify <code>/user/betty/\${filename}</code>, the key name is <code>/user/betty/lolcatz.jpg</code>.</p> <p>For more information, go to <a href="#">Object Key and Metadata</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p>	Yes

Name	Description	Required
policy	<p>Security Policy describing what is permitted in the request. Requests without a security policy are considered anonymous and work only on publicly writable buckets. For more information, go to <a href="#">HTML Forms and Upload Examples</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Policy is required if the bucket is not publicly writable.</p>	Conditional
success_action_redirect, redirect	<p>The URL to which the client is redirected upon successful upload.</p> <p>If <code>success_action_redirect</code> is not specified, Amazon S3 returns the empty document type specified in the <code>success_action_status</code> field.</p> <p>If Amazon S3 cannot interpret the URL, it acts as if the field is not present.</p> <p>If the upload fails, Amazon S3 displays an error and does not redirect the user to a URL.</p> <p>Type: String</p> <p>Default: None</p> <p><b>Note</b> The redirect field name is deprecated, and support for the redirect field name is removed in the future.</p>	No

Name	Description	Required
success_action_status	<p>If you don't specify <code>success_action_redirect</code>, the status code is returned to the client when the upload succeeds.</p> <p>Accepts the values 200, 201, or 204 (the default).</p> <p>If the value is set to 200 or 204, Amazon S3 returns an empty document with a 200 or 204 status code.</p> <p>If the value is set to 201, Amazon S3 returns an XML document with a 201 status code.</p> <p>If the value is not set or if it is set to an invalid value, Amazon S3 returns an empty document with a 204 status code.</p> <p>Type: String</p> <p>Default: None</p> <p><b>Note</b> Some versions of the Adobe Flash player do not properly handle HTTP responses with an empty body. To support uploads through Adobe Flash, we recommend setting <code>success_action_status</code> to 201.</p>	No
tagging	<p>Specifies set of tags to add to the object using the following encoding scheme.</p> <pre>&lt;Tagging&gt;   &lt;TagSet&gt;     &lt;Tag&gt;       &lt;Key&gt;Tag Name&lt;/Key&gt;       &lt;Value&gt;Tag Value&lt;/Value&gt;     &lt;/Tag&gt;     ...   &lt;/TagSet&gt; &lt;/Tagging&gt;</pre> <p>For more information, see <a href="#">Object Tagging</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p>	No

Name	Description	Required
x-amz-storage-class	<p>Storage class to use for storing the object. If you don't specify a class, Amazon S3 uses the default storage class, STANDARD. Amazon S3 supports other storage classes. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: STANDARD</p> <p>Valid Values: STANDARD   STANDARD_IA   ONEZONE_IA   INTELLIGENT_TIERING   GLACIER   REDUCED_REDUNDANCY</p>	No
x-amz-meta-*	<p>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see <a href="#">PUT Object (p. 412)</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
x-amz-security-token	<p>Amazon DevPay security token.</p> <p>Each request that uses Amazon DevPay requires two x-amz-security-token form fields: one for the product token and one for the user token.</p> <p>Type: String</p> <p>Default: None</p>	No

Name	Description	Required
x-amz-website-redirect-location	<p>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see <a href="#">Object Key and Metadata</a>.</p> <p>In the following example, the request header sets the redirect to an object (<code>anotherPage.html</code>) in the same bucket:</p> <pre>x-amz-website-redirect-location: /anotherPage.html</pre> <p>In the following example, the request header sets the object redirect to another website:</p> <pre>x-amz-website-redirect-location: http://www.example.com/</pre> <p>For more information about website hosting in Amazon S3, see <a href="#">Hosting Websites on Amazon S3</a> and <a href="#">How to Configure Website Page Redirects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The value must be prefixed by, "/", "http://" or "https://". The length of the value is limited to 2 K.</p>	No

## Server-Side Encryption Specific Request Form Fields

You can optionally request Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it.

For more information, see [Protecting Data Using Server-Side Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

Depending on whether you want to use AWS-managed encryption keys or provide your own encryption keys, the following form fields:

- Use AWS-managed encryption keys — If you want Amazon S3 to manage keys used to encrypt data, specify the following form fields in the request.

Name	Description	Required
x-amz-server-side-encryption	<p>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</p> <p>Type: String</p>	Yes

Name	Description	Required
	Valid Value: aws:kms, AES256	
x-amz-server-side-encryption-aws-kms-key-id	If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.  Type: String	Yes, if the value of x-amz-server-side-encryption is aws:kms
x-amz-server-side-encryption-context	If x-amz-server-side-encryption is present, and if its value is aws:kms, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the key-value pairs for the encryption context.  Type: String	No

**Note**

If you specify x-amz-server-side-encryption:aws:kms, but do not provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

- Use customer-provided encryption keys — If you want to manage your own encryption keys, you must provide all the following form fields in the request.

**Note**

If you use this feature, the ETag value that Amazon S3 returns in the response is not the MD5 of the object.

Name	Description	Required
x-amz-server-side-encryption-customer-algorithm	Specifies the algorithm to use to when encrypting the object.  Type: String  Default: None  Valid Value: AES256  Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 fields.	Yes
x-amz-server-side-encryption-customer-key	Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then it is discarded. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.  Type: String  Default: None	Yes

Name	Description	Required
	Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> fields.	
<code>x-amz-server-side-encryption-customer-key-MD5</code>	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> fields.</p>	Yes

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
<code>x-amz-expiration</code>	If an <code>Expiration</code> action is configured for the object as part of the bucket's lifecycle configuration, Amazon S3 returns this header. The header value includes an "expiry-date" component and a URL-encoded "rule-id" component. For version-enabled buckets, this header applies only to current versions. Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see <a href="#">PUT Bucket lifecycle (p. 265)</a> .
	Type: String
<code>success_action_redirect</code> , <code>redirect</code>	The URL to which the client is redirected on successful upload.
	Type: String
	Ancestor: PostResponse
<code>x-amz-server-side-encryption</code>	If you specified server-side encryption either with AWS KMS encryption or AWS-managed encryption in your POST request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.
	Type: String
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	If the <code>x-amz-server-side-encryption</code> header is present and has the value of <code>aws:kms</code> , this header

Name	Description
	<p>specifies the ID of the AWS KMS master encryption key that was used for the object.</p> <p>Type: String</p>
x-amz-server-side-encryption-customer-algorithm	<p>If server-side encryption with customer-provided encryption keys (SSE-C) encryption was requested, the response includes this header that confirms the encryption algorithm that was used.</p> <p>Type: String</p> <p>Valid Values: AES256</p>
x-amz-server-side-encryption-customer-key-MD5	<p>If SSE-C encryption was requested, the response includes this header to verify roundtrip message integrity of the customer-provided encryption key.</p> <p>Type: String</p>
x-amz-version-id	<p>Version of the object.</p> <p>Type: String</p>

## Response Elements

Name	Description
Bucket	<p>Name of the bucket the object was stored in.</p> <p>Type: String</p> <p>Ancestor: PostResponse</p>
ETag	<p>The entity tag is an MD5 hash of the object that you can use to do conditional GET operations using the If-Modified request tag with the GET request operation. ETag reflects changes only to the contents of an object, not its metadata.</p> <p>Type: String</p> <p>Ancestor: PostResponse</p>
Key	<p>The object key name.</p> <p>Type: String</p> <p>Ancestor: PostResponse</p>
Location	<p>URI of the object.</p> <p>Type: String</p> <p>Ancestor: PostResponse</p>

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

```
POST /Neo HTTP/1.1
Content-Length: 4
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2006 12:00:00 GMT
Authorization: authorization string
Content-Type: text/plain
Expect: the 100-continue HTTP status code

ObjectContent
```

### Sample Response with Versioning Suspended

The following is a sample response when bucket versioning is suspended:

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiiIgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: default
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

In this response, the version ID is null.

### Sample Response with Versioning Enabled

The following is a sample response when bucket versioning is enabled.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiiIgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fjj3HHNVfdsQUIFDNsidf038jfdsjGFDSIRp
Date: Wed, 01 Mar 2006 12:00:00 GMT
ETag: "828ef3fd9a96f00ad9f27c383fc9ac7f"
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Object - Copy \(p. 431\)](#)
- [POST Object \(p. 385\)](#)
- [GET Object \(p. 349\)](#)



# POST Object restore

## Description

This operation performs the following types of requests:

- `select` – Perform a select query on an archived object
- `restore an archive` – Restore an archived object

To use this operation, you must have permissions to perform the `s3:RestoreObject` and `s3:GetObject` actions. The bucket owner has this permission by default and can grant this permission to others. For more information about permissions, see [Permissions Related to Bucket Subresource Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Querying Archives with Select Requests

You use a select type of request to perform SQL queries on archived objects. The archived objects that are being queried by the select request must be formatted as uncompressed comma-separated values (CSV) files. You can run queries and custom analytics on your archived data without having to restore your data to a hotter Amazon S3 tier. For an overview about select requests, see [Querying Archived Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

When making a select request, do the following:

- Define an output location for the select query's output. This must be an Amazon S3 bucket in the same AWS Region as the bucket that contains the archive object that is being queried. The AWS account that initiates the job must have permissions to write to the S3 bucket. You can specify the storage class and encryption for the output objects stored in the bucket. For more information about output, see [Querying Archived Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

For more information about the S3 structure in the request body, see the following:

- [PUT Object \(p. 412\)](#)
- [Managing Access with ACLs in the Amazon Simple Storage Service Developer Guide](#)
- [Protecting Data Using Server-Side Encryption in the Amazon Simple Storage Service Developer Guide](#)
- Define the SQL expression for the `SELECT` type of restoration for your query in the request body's `SelectParameters` structure. You can use expressions like the following examples.
- The following expression returns all records from the specified object.

```
SELECT * FROM Object
```

- Assuming that you are not using any headers for data stored in the object, you can specify columns with positional headers.

```
SELECT s._1, s._2 FROM Object s WHERE s._3 > 100
```

- If you have headers and you set the `fileHeaderInfo` in the CSV structure in the request body to `USE`, you can specify headers in the query. (If you set the `fileHeaderInfo` field to `IGNORE`, the first row is skipped for the query.) You cannot mix ordinal positions with header column names.

```
SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
```

For more information about using SQL with Glacier Select restore, see [SQL Reference for Amazon S3 Select and Glacier Select](#) in the *Amazon Simple Storage Service Developer Guide*.

When making a select request, you can also do the following:

- To expedite your queries, specify the `Expedited` tier. For more information about tiers, see "Restoring Archives," later in this topic.
- Specify details about the data serialization format of both the input object that is being queried and the serialization of the CSV-encoded query results.

The following are additional important facts about the select feature:

- The output results are new Amazon S3 objects. Unlike archive retrievals, they are stored until explicitly deleted—manually or through a lifecycle policy.
- You can issue more than one select request on the same Amazon S3 object. Amazon S3 doesn't deduplicate requests, so avoid issuing duplicate requests.
- Amazon S3 accepts a select request even if the object has already been restored. A select request doesn't return error response 409.

## Restoring Archives

The restore request restores a temporary copy of an archived object. To restore a specific object version, you can provide a version ID. If you don't provide a version ID, Amazon S3 restores the current version.

Objects in the `GLACIER` storage class are archived. To access an archived object, you must first initiate a restore request. This restores a copy of the archived object. The time it takes restore jobs to finish depends on which data access tier you specify, `Expedited`, `Standard`, or `Bulk`.

In a restore request, you specify the number of days that you want the restored copy to exist. After the specified period, Amazon S3 deletes the temporary copy. The object remains archived. Amazon S3 deletes only the restored copy.

When restoring an archived object (or using a select request), you can specify one of the following options in the `Tier` element of the request body:

- **Expedited** – Lets you quickly access your data when occasional urgent requests for a subset of archives are required. For all but the largest archived object (250 MB+), data accessed using Expedited retrievals are typically made available within 1–5 minutes.
- **Standard** – Lets you access any of your archived objects within several hours. Standard retrievals typically finish within 3–5 hours. This is the default tier.
- **Bulk** – The lowest-cost data access option in Glacier. It lets you retrieve large amounts, even petabytes, of data inexpensively in a day. Bulk access typically completes within 5–12 hours.

For more information about archive retrieval options and provisioned capacity for `Expedited` data access, see [Restoring Archived Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

You can use Amazon S3 restore speed upgrade to change the restore speed to a faster speed while it is in progress. You upgrade the speed of an in-progress restoration by issuing another restore request to the same object, setting a new `Tier` request element. When issuing a request to upgrade the restore tier, you must choose a tier that is faster than the tier that the in-progress restore is using. You must not change any other parameters, such as the `Days` request element. For more information, see [Upgrading the Speed of an In-Progress Restore](#) in the *Amazon Simple Storage Service Developer Guide*.

To get the status of object restoration, you can send a `HEAD` request. Operations return the `x-amz-restore` header, which provides information about the restoration status, in the response. You can

use Amazon S3 event notifications to notify you when a restore is initiated or completed. For more information, see [Configuring Amazon S3 Event Notifications](#) in the *Amazon Simple Storage Service Developer Guide*.

After restoring an archived object, you can update the restoration period by reissuing the request with a new period. Amazon S3 updates the restoration period relative to the current time and charges only for the request—there are no data transfer charges.

You cannot issue another restore request for an object when Amazon S3 is actively processing your first restore request for the same object. However, after Amazon S3 restores a copy of the object, you can send restore requests to update the expiration period of the restored object copy.

If your bucket has a lifecycle configuration with a rule that includes an expiration action, the object expiration overrides the life span that you specify in a restore request. For example, if you restore an object copy for 10 days, but the object is scheduled to expire in 3 days, Amazon S3 deletes the object in 3 days. For more information about lifecycle configuration, see [PUT Bucket lifecycle \(p. 265\)](#) and [Object Lifecycle Management](#) in *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
POST /ObjectName?restore&versionId=VersionID HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Content-MD5: MD5

request body
```

#### Note

The syntax shows some of the request headers. For a complete list, see "Request Headers," later in this topic.

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message integrity check to verify that the request body was not corrupted in transit. For more information, see <a href="#">RFC 1864</a> .  Type: String  Default: None	Yes

### Request Elements

The following is an XML example of a request body for restoring an archive.

```
<RestoreRequest>
  <Days>2</Days>
  <GlacierJobParameters>
    <Tier>Bulk</Tier>
  </GlacierJobParameters>
</RestoreRequest>
```

The following table explains the XML for archive restoration in the request body.

Name	Description	Required
RestoreRequest	Container for restore information.  Type: Container	Yes
Days	Lifetime of the restored (active) copy. The minimum number of days that you can restore an object from Glacier is 1. After the object copy reaches the specified lifetime, Amazon S3 removes it from the bucket. If you are restoring an archive, this element is required.  Do not use this element with a <code>SELECT</code> type of request.  Type: Positive integer  Ancestors: <code>RestoreRequest</code>	Yes, if restoring an archive
GlacierJobParameters	Container for Glacier job parameters.  Do not use this element with a <code>SELECT</code> type of request.  Type: Container  Ancestors: <code>RestoreRequest</code>	No
Tier	The data access tier to use when restoring the archive. <code>Standard</code> is the default.  Type: Enum  Valid values: <code>Expedited</code>   <code>Standard</code>   <code>Bulk</code>  Ancestors: <code>GlacierJobParameters</code>	No

The following XML is the request body for a select query on an archived object:

```
<RestoreRequest>
  <Type>SELECT</Type>
  <Tier>Expedited</Tier>
  <Description>Job description</Description>
  <SelectParameters>
    <Expression>Select * from Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
      <CSV>
        <FileHeaderInfo>IGNORE</FileHeaderInfo>
        <RecordDelimiter>\n</RecordDelimiter>
        <FieldDelimiter>,</FieldDelimiter>
        <QuoteCharacter>"</QuoteCharacter>
        <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
```

```

        <Comments>#</Comments>
    </CSV>
</InputSerialization>
<OutputSerialization>
    <CSV>
        <QuoteFields>ASNEEDED</QuoteFields>
        <RecordDelimiter>\n</RecordDelimiter>
        <FieldDelimiter>,</FieldDelimiter>
        <QuoteCharacter>"</QuoteCharacter>
        <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
    </CSV>
</OutputSerialization>
</SelectParameters>
<OutputLocation>
    <S3>
        <BucketName>Name of bucket</BucketName>
        <Prefix>Key prefix</Prefix>
        <CannedACL>Canned ACL string</CannedACL>
        <AccessControlList>
            <Grantee>
                <Type>Grantee Type</Type>
            <ID>Grantee identifier</ID>
            <URI>Grantee URI</URI>
                <Permission>Granted permission</Permission>
                    <DisplayName>Display Name</DisplayName>
                    <EmailAddress>email</EmailAddress>
            </Grantee>
        </AccessControlList>
        <Encryption>
            <EncryptionType>Encryption type</EncryptionType>
        <KMSKeyId>KMS Key ID</KMSKeyId>
        <KMSCluster>Base64-encoded JSON<KMSCluster>
            </Encryption>
        <UserMetadata>
            <MetadataEntry>
                <Name>Key</Name>
                <Value>Value</Value>
            </MetadataEntry>
        </UserMetadata>
        <Tagging>
            <TagSet>
                <Tag>
                    <Key>Tag name</Key>
                    <Value>Tag value</Value>
                </Tag>
            </TagSet>
        </Tagging>
        <StorageClass>Storage class</StorageClass>
    </S3>
</OutputLocation>
</RestoreRequest>

```

The following tables explain the XML for a **SELECT** type of restoration in the request body.

Name	Description	Required
RestoreRequest	Container for restore information.  Type: Container	Yes
Tier	The data access tier to use when restoring the archive.  Standard is the default.  Type: Enum	No

Name	Description	Required
	Valid values: Expedited   Standard   Bulk  Ancestors: <code>RestoreRequest</code>	
<code>Description</code>	The optional description for the request.  Type: String  Ancestors: <code>RestoreRequest</code>	No
<code>SelectParameters</code>	Describes the parameters for the select job request.  Type: Container  Ancestors: <code>RestoreRequest</code>	Yes, if request type is <code>SELECT</code>
<code>OutputLocation</code>	Describes the location that receives the results of the select restore request.  Type: Container for Amazon S3  Ancestors: <code>RestoreRequest</code>	Yes, if request type is <code>SELECT</code>

**The `SelectParameters` container element contains the following elements.**

Name	Description	Required
<code>Expression</code>	The SQL expression. For example: <ul style="list-style-type: none"><li>• The following SQL expression retrieves the first column of the data from the object stored in CSV format:  <code>SELECT s._1 FROM Object s</code></li><li>• The following SQL expression returns everything from the object:  <code>SELECT * FROM Object</code></li></ul> Type: String  Ancestors: <code>SelectParameters</code>	Yes
<code>ExpressionType</code>	Identifies the expression type.  Type: String  Valid values: SQL  Ancestors: <code>SelectParameters</code>	Yes
<code>InputSerializer</code>	Describes the serialization format of the object.  Type: Container for CSV  Ancestors: <code>SelectParameters</code>	Yes
<code>OutputSerializer</code>	Describes how the results of the select job are serialized.	Yes

Name	Description	Required
	Type: Container for CSV  Ancestors: SelectParameters	

**The CSV container element in the `InputSerialization` element contains the following elements.**

Name	Description	Required
RecordDelimiter	A single character used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter.  Type: String  Default: \n  Ancestors: CSV	No
FieldDelimiter	A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.  Type: String  Default: ,  Ancestors: CSV	No
QuoteCharacter	A single character used for escaping when the field delimiter is part of the value.  Consider this example in a CSV file:  "a, b"  Wrapping the value in quotation marks makes this value a single field. If you don't use the quotation marks, the comma is a field delimiter (which makes it two separate field values, a and b).  Type: String  Default: "  Ancestors: CSV	No
QuoteEscapeChar	A single character used for escaping the quotation mark character inside an already escaped value. For example, the value """ a , b """ is parsed as " a , b ".  Type: String  Default: "  Ancestors: CSV	No
FileInfo	Describes the first line in the input data. It is one of the ENUM values.	No

Name	Description	Required
	<ul style="list-style-type: none"> <li>• <b>NONE</b>: First line is not a header.</li> <li>• <b>IGNORE</b>: First line is a header, but you can't use the header values to indicate the column in an expression. You can use column position (such as <code>_1</code>, <code>_2</code>, ...) to indicate the column (<code>SELECT s._1 FROM OBJECT s</code>).</li> <li>• <b>USE</b>: First line is a header, and you can use the header value to identify a column in an expression (<code>SELECT "name" FROM OBJECT</code>).</li> </ul> <p>Type: Enum Valid values: <code>NONE</code>   <code>USE</code>   <code>IGNORE</code> Ancestors: CSV</p>	
Comments	<p>A single character used to indicate that a row should be ignored when the character is present at the start of that row. You can specify any character to indicate a comment line.</p> <p>Type: String Ancestors: CSV</p>	No

**The CSV container element (in the `OutputSerialization` elements) contains the following elements.**

Name	Description	Required
QuoteFields	<p>Indicates whether to use quotation marks around output fields.</p> <ul style="list-style-type: none"> <li>• <b>ALWAYS</b>: Always use quotation marks for output fields.</li> <li>• <b>ASNEEDED</b>: Use quotation marks for output fields when needed.</li> </ul> <p>Type: Enum Valid values: <code>ALWAYS</code>   <code>ASNEEDED</code> Default: <code>AsNeeded</code> Ancestors: CSV</p>	No
RecordDelimiter	<p>A single character used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter.</p> <p>Type: String Default: <code>\n</code> Ancestors: CSV</p>	No
FieldDelimiter	<p>A single character used to separate individual fields in a record. You can specify an arbitrary delimiter.</p>	No

Name	Description	Required
	<p>Type: String</p> <p>Default: ,</p> <p>Ancestors: CSV</p>	
QuoteCharacter	<p>A single character used for escaping when the field delimiter is part of the value. For example, if the value is a, b, Amazon S3 wraps this field value in quotation marks, as follows: " a , b ".</p> <p>Type: String</p> <p>Default: "</p> <p>Ancestors: CSV</p>	No
QuoteEscapeCharacter	<p>A single character used for escaping the quotation mark character inside an already escaped value. For example, if the value is " a , b ", Amazon S3 wraps the value in quotation marks, as follows: """ a , b """.</p> <p>Type: String</p> <p>Ancestors: CSV</p>	No

**The S3 container element (in the OutputLocation element) contains the following elements.**

Name	Description	Required
AccessControlList	<p>A list of grants that control access to the staged results.</p> <p>Type: Container for Grant</p> <p>Ancestors: S3</p>	No
BucketName	<p>The name of the S3 bucket where the select restore results are stored. The bucket must be in the same AWS Region as the bucket that contains the input archive object.</p> <p>Type: String</p> <p>Ancestors: S3</p>	Yes
CannedACL	<p>The canned access control list (ACL) to apply to the select restore results.</p> <p>Type: String</p> <p>Valid values: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</p> <p>Ancestors: S3</p>	No
Encryption	Contains encryption information for the stored results.	No

Name	Description	Required
	Type: Container for Encryption  Ancestors: S3	
Prefix	The prefix that is prepended to the select restore results. The maximum length for the prefix is 512 bytes.  Type: String  Ancestors: S3	Yes
StorageClass	The class of storage used to store the select request results.  Type: String  Valid values: STANDARD   REDUCED_REDUNDANCY   STANDARD_IA   ONEZONE_IA  Ancestors: S3	No
Tagging	Container for tag information.  Type: Tag structure  Ancestors: S3	No
UserMetadata	Contains a list of metadata to store with the select restore results.  Type: MetadataEntry structure  Ancestors: S3	No

**The Grantee container element (in the AccessControlList element) contains the following elements.**

Name	Description	Required
DisplayName	The screen name of the grantee.  Type: String  Ancestors: Grantee	No
EmailAddress	The email address of the grantee.  Type: String  Ancestors: Grantee	No
ID	The canonical user ID of the grantee.  Type: String  Ancestors: Grantee	No
Type	The type of the grantee.  Type: String	No

Name	Description	Required
	Ancestors: Grantee	
URI	The URI of the grantee group.  Type: String  Ancestors: Grantee	No
Permission	Granted permission.  Type: String  Ancestors: Grantee	No

**The `Encryption` container element (in S3) contains the following elements.**

Name	Description	Required
EncryptionType	The server-side encryption algorithm used when storing job results. The default is no encryption.  Type: String  Valid Values <code>aws:kms</code>   <code>AES256</code>  Ancestors: <code>Encryption</code>	No
KMSContext	Optional. If the encryption type is <code>aws:kms</code> , you can use this value to specify the encryption context for the select restore results.  Type: String  Ancestors: <code>Encryption</code>	No
KMSKeyId	The AWS Key Management Service (AWS KMS) key ID to use for object encryption.  Type: String  Ancestors: <code>Encryption</code>	No

**The `TagSet` container element (in the Tagging element) contains the following element.**

Name	Description	Required
Tag	Contains tags.  Type: Container  Ancestors: <code>TagSet</code>	No

The Tag container element (in the TagSet element) contains the following elements.

Name	Description	Required
Key	Name of the tag.  Type: String  Ancestors: Tag	No
Value	Value of the tag.  Type: String  Ancestors: Tag	No

The MetadataEntry container element (in the UserMetadata element) contains the following key-value pair elements to store with an object.

Name	Description	Required
MetadataKey	The metadata key.  Type: String  Ancestors:	No
MetadataEntry	The metadata value.  Type: String  Ancestors:	No

## Responses

A successful operation returns either the 200 OK or 202 Accepted status code.

- If the object copy is not previously restored, then Amazon S3 returns 202 Accepted in the response.
- If the object copy is previously restored, Amazon S3 returns 200 OK in the response.

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Elements

This operation does not return response elements.

## Special Errors

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
RestoreAlreadyInProgress	Object restore is already in progress. (This error does not apply to SELECT type requests.)	409 Conflict	Client
GlacierExpeditedRetrievalNotAvailable	Glacier expedited retrievals are currently not available. Try again later. (Returned if there is insufficient capacity to process the Expedited request. This error applies only to Expedited retrievals and not to Standard or Bulk retrievals.)	503	N/A

## Examples

### Restore an Object for Two Days Using the Expedited Retrieval Option

The following restore request restores a copy of the `photo1.jpg` object from Glacier for a period of two days using the expedited retrieval option.

```
POST /photo1.jpg?restore HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Mon, 22 Oct 2012 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<RestoreRequest>
  <Days>2</Days>
  <GlacierJobParameters>
    <Tier>Expedited</Tier>
  </GlacierJobParameters>
</RestoreRequest>
```

If the `examplebucket` does not have a restored copy of the object, Amazon S3 returns the following 202 Accepted response.

```
HTTP/1.1 202 Accepted
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/cHR3Yb2fCb2S04nxI423Dqwg2XiQ0B/UZlzYQvPiBlZNRCovw=
x-amz-request-id: 9F341CD3C4BA79E0
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3
```

If a copy of the object is already restored, Amazon S3 returns a 200 OK response, and updates only the restored copy's expiry time.

### Query an Archive with a SELECT Request

The following is an example select restore request.

```
POST /object-one.csv?restore HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Date: Sat, 20 Oct 2012 23:54:05 GMT
Authorization: authorization string
Content-Length: content length

<RestoreRequest xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Type>SELECT</Type>
  <Tier>Expedited</Tier>
  <Description>this is a description</Description>
  <SelectParameters>
    <InputSerialization>
      <CSV>
        <FileHeaderInfo>IGNORE</FileHeaderInfo>
        <Comments>#</Comments>
        <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        <RecordDelimiter>\n</RecordDelimiter>
        <FieldDelimiter>,</FieldDelimiter>
        <QuoteCharacter>"</QuoteCharacter>
      </CSV>
    </InputSerialization>
    <ExpressionType>SQL</ExpressionType>
    <Expression>select * from object</Expression>
  <OutputSerialization>
    <CSV>
      <QuoteFields>ALWAYS</QuoteFields>
      <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
      <RecordDelimiter>\n</RecordDelimiter>
      <FieldDelimiter>\t</FieldDelimiter>
      <QuoteCharacter>\ '</QuoteCharacter>
    </CSV>
  </OutputSerialization>
</SelectParameters>
<OutputLocation>
  <S3>
    <BucketName>example-output-bucket</BucketName>
    <Prefix>test-s3</Prefix>
    <AccessControlList>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AmazonCustomerByEmail">
          <EmailAddress>jane-doe@example.com</EmailAddress>
        </Grantee>
        <Permission>FULL_CONTROL</Permission>
      </Grant>
    </AccessControlList>
    <UserMetadata>
      <MetadataEntry>
        <Name>test</Name>
        <Value>test-value</Value>
      </MetadataEntry>
      <MetadataEntry>
        <Name>other</Name>
        <Value>something else</Value>
      </MetadataEntry>
    </UserMetadata>
    <StorageClass>STANDARD</StorageClass>
  </S3>
</OutputLocation>
</RestoreRequest>
```

Amazon S3 returns the following 202 Accepted response.

```
HTTP/1.1 202 Accepted
```

```
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/cHR3Yb2fCb2S04nxI423Dqwg2XiQ0B/UZlzYQvPiBlZNRCovw=
x-amz-request-id: 9F341CD3C4BA79E0
x-amz-restore-output-path: js-test-s3/qE8nk5MOXIj-LuZE2HXNw6empQm3znLkHlMWInRYPs-
Orl2W0uj6LyYm-neTvm1-btz3wbBxfMhPykd3jkl-lvZE7w42/
Date: Sat, 20 Oct 2012 23:54:05 GMT
Content-Length: 0
Server: AmazonS3
```

## More Info

- [GET Bucket lifecycle \(p. 145\)](#)
- [PUT Bucket lifecycle \(p. 265\)](#)
- [SQL Reference for Amazon S3 Select and Glacier Select](#) in the *Amazon Simple Storage Service Developer Guide*

# PUT Object

## Description

This implementation of the `PUT` operation adds an object to a bucket. You must have `WRITE` permissions on a bucket to add an object to it.

Amazon S3 never adds partial objects; if you receive a success response, Amazon S3 added the entire object to the bucket.

Amazon S3 is a distributed system. If it receives multiple write requests for the same object simultaneously, it overwrites all but the last object written. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer or use versioning instead.

To ensure that data is not corrupted traversing the network, use the `Content-MD5` header. When you use this header, Amazon S3 checks the object against the provided MD5 value and, if they do not match, returns an error. Additionally, you can calculate the MD5 while putting an object to Amazon S3 and compare the returned ETag to the calculated MD5 value.

### Note

To configure your application to send the request headers before sending the request body, use the `100-continue` HTTP status code. For `PUT` operations, this helps you avoid sending the message body if the message is rejected based on the headers (for example, because authentication fails or a redirect occurs). For more information on the `100-continue` HTTP status code, go to Section 8.2.3 of <http://www.ietf.org/rfc/rfc2616.txt>.

You can optionally request server-side encryption. With server-side encryption, Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts the data when you access it. You have the option to provide your own encryption key or use AWS-managed encryption keys. For more information, see [Using Server-Side Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

## Versioning

If you enable versioning for a bucket, Amazon S3 automatically generates a unique version ID for the object being stored. Amazon S3 returns this ID in the response using the `x-amz-version-id` response header. If versioning is suspended, Amazon S3 always uses `null` as the version ID for the object stored. For more information about returning the versioning state of a bucket, see [GET Bucket versioning \(p. 199\)](#).

If you enable versioning for a bucket, when Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of the objects.

To see sample requests that use versioning, see [Sample Request \(p. 423\)](#).

## Storage Class Options

By default, Amazon S3 uses the Standard storage class to store newly created objects. The Standard storage class provides high durability and high availability. You can specify other storage classes depending on the performance needs. For more information, see [Storage Classes](#) in the *Amazon Simple Storage Service Developer Guide*.

## Access Permissions

When uploading an object, you can optionally specify the accounts or groups that should be granted specific permissions on your object. There are two ways to grant the appropriate permissions using the request headers:

- Specify a canned (predefined) ACL using the `x-amz-acl` request header. For more information, see [Canned ACL](#) in the *Amazon Simple Storage Service Developer Guide*.
- Specify access permissions explicitly using the `x-amz-grant-read`, `x-amz-grant-read-acp`, and `x-amz-grant-write-acp`, `x-amz-grant-full-control` headers. These headers map to the set of permissions Amazon S3 supports in an ACL. For more information, go to [Access Control List \(ACL\) Overview](#) in the *Amazon Simple Storage Service Developer Guide*.

**Note**

You can either use a canned ACL or specify access permissions explicitly. You cannot do both.

To change an object's ACLs from the default, the requester must have `s3:PutObjectAcl` included in the list of permitted actions in their AWS Identity and Access Management (IAM) policy. For more information about permissions, see [Permissions for Object Operations](#) and [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

**Note**

The syntax shows some of the request headers. For a complete list, see the Request Headers section.

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
Cache-Control	<p>Can be used to specify caching behavior along the request/reply chain. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
Content-Disposition	<p>Specifies presentational information for the object. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1">http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1</a>.</p> <p>Type: String</p>	No

Name	Description	Required
	<p>Default: None</p> <p>Constraints: None</p>	
Content-Encoding	<p>Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
Content-Length	<p>The size of the object, in bytes. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	Yes
Content-MD5	<p>The base64-encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, see <a href="#">REST Authentication</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
Content-Type	<p>A standard MIME type describing the format of the contents. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17</a>.</p> <p>Type: String</p> <p>Default: binary/octet-stream</p> <p>Valid Values: MIME types</p> <p>Constraints: None</p>	No

Name	Description	Required
Expect	<p>When your application uses <code>100-continue</code>, it does not send the request body until it receives an acknowledgment. If the message is rejected based on the headers, the body of the message is not sent.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Values: <code>100-continue</code></p> <p>Constraints: None</p>	No
Expires	<p>The date and time at which the object is no longer able to be cached. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-meta-	<p>Headers starting with this prefix are user-defined metadata. Within the PUT request header, the user-defined metadata is limited to 2 KB in size. User-defined metadata is a set of key-value pairs. The size of user-defined metadata is the sum of the number of bytes in the UTF-8 encoding of each key and value. Amazon S3 doesn't validate or interpret user-defined metadata.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-storage-class	<p>If you don't specify, Standard is the default storage class. Amazon S3 supports other storage classes. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Enum</p> <p>Default: STANDARD</p> <p>Valid Values: STANDARD   STANDARD_IA   ONEZONE_IA   INTELLIGENT_TIERING   GLACIER   REDUCED_REDUNDANCY</p>	No

Name	Description	Required
x-amz-tagging	<p>Specifies a set of one or more tags to associate with the object. These tags are stored in the tagging subresource that is associated with the object.</p> <p>To specify tags on an object, the requester must have <code>s3:PutObjectTagging</code> included in the list of permitted actions in their IAM policy.</p> <p>For more information about adding tags to an object, see <a href="#">Object Tagging Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The encoding for tags is URL query parameter encoding. The maximum size of this header is 2 KB.</p>	No
x-amz-website-redirect-location	<p>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see <a href="#">Object Key and Metadata</a>.</p> <p>In the following example, the request header sets the redirect to an object (<code>anotherPage.html</code>) in the same bucket:</p> <pre>x-amz-website-redirect-location: /anotherPage.html</pre> <p>In the following example, the request header sets the object redirect to another website:</p> <pre>x-amz-website-redirect-location: http://www.example.com/</pre> <p>For more information about website hosting in Amazon S3, see <a href="#">Hosting Websites on Amazon S3</a> and <a href="#">How to Configure Website Page Redirects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The value must be prefixed by, "/", "http://" or "https://". The length of the value is limited to 2 KB.</p>	No

Name	Description	Required
x-amz-object-lock-mode	<p>The Object Lock mode, if any, that should be applied to this object. For more information about S3 Object Lock, see <a href="#">Object Lock</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid values: GOVERNANCE   COMPLIANCE</p>	No
x-amz-object-lock-retain-until-date	<p>The date and time when the Object Lock retention period will expire.</p> <p>Type: Timestamp</p> <p>Default: None</p> <p>Format: <code>2020-01-05T00:00:00.000Z</code></p>	Required if <code>x-amz-object-lock-mode</code> is specified
x-amz-object-lock-legal-hold	<p>Specifies whether a legal hold will be applied to this object. For more information about legal holds, see <a href="#">Object Lock</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid values: ON   OFF</p>	No

## Access-Control-List-(ACL)-Specific Request Headers

Additionally, you can use the following access control-related headers with this operation. By default, all objects are private: only the owner has full control. When adding a new object, you can grant permissions to individual AWS accounts or predefined Amazon S3 groups. These permissions are then used to create the Access Control List (ACL) on the object. For more information, see [Using ACLs](#).

To grant these permissions, you can use one of the following methods:

- **Specify a canned ACL** — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, go to [Canned ACL](#).

Name	Description	Required
x-amz-acl	<p>The canned ACL to apply to the object. For more information, see <a href="#">Canned ACL</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: private</p> <p>Valid Values: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</p>	No

Name	Description	Required
	Constraints: None	

- **Specify access permissions explicitly** — To explicitly grant access permissions to specific AWS accounts or a group, use the following headers. Each maps to specific permissions that Amazon S3 supports in an ACL. For more information, see [Access Control List \(ACL\) Overview](#). In the header value, you specify a list of grantees who get the specific permission.

Name	Description	Required
x-amz-grant-read	Grants permission to read the object data and its metadata.  Type: String  Default: None  Constraints: None	No
x-amz-grant-write	Not applicable. This header applies only when granting permission on a bucket.  Type: String  Default: None  Constraints: None	No
x-amz-grant-read-acp	Grants permission to read the object ACL.  Type: String  Default: None  Constraints: None	No
x-amz-grant-write-acp	Grants permission to write the ACL for the applicable object.  Type: String  Default: None  Constraints: None	No
x-amz-grant-full-control	Grants READ, READ_ACP, and WRITE_ACP permissions on the object.  Type: String  Default: None  Constraints: None	No

You specify each grantee as a `type=value` pair, where the type can be one of the following:

- **emailAddress** – if the specified value is the email address of an AWS account

### **Important**

Using email addresses to specify a grantee is only supported in the following AWS Regions:

- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- EU (Ireland)
- South America (São Paulo)

For a list of all the Amazon S3 supported regions and endpoints, see [Regions and Endpoints](#) in the [AWS General Reference](#).

- **id** – if the specified value is the canonical user ID of an AWS account
- **uri** – if you are granting permission to a predefined group

For example, the following `x-amz-grant-read` header grants permission to read object data and its metadata to the AWS accounts identified by their email addresses.

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

### [Server-Side-Encryption-Specific Request Headers](#)

You can optionally request Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts the data when you access it. The header you use depend on whether you want to use AWS-managed encryption keys or provide your own encryption keys.

- Use AWS-managed encryption keys — If you want Amazon S3 to manage the keys used to encrypt data, specify the following headers in the request.

Name	Description	Required
<code>x-amz-server-side-encryption</code>	<p>Specifies the server-side encryption algorithm to use when Amazon S3 creates an object.</p> <p>Type: String</p> <p>Valid Value: <code>aws:kms</code>, <code>AES256</code></p>	Yes
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	<p>If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code>, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.</p> <p>Type: String</p>	Yes, if the value of <code>x-amz-server-side-encryption</code> is <code>aws:kms</code>
<code>x-amz-server-side-encryption-context</code>	<p>If the <code>x-amz-server-side-encryption</code> header is present, and if its value is <code>aws:kms</code>, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.</p>	No

Name	Description	Required
	Type: String	

**Note**

If you specify `x-amz-server-side-encryption:aws:kms`, but do not provide `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 uses the default AWS KMS key to protect the data.

**Important**

All GET and PUT requests for an object protected by AWS KMS fail if you don't make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see [Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys](#) in the *Amazon Simple Storage Service Developer Guide*.

- Use customer-provided encryption keys— If you want to manage your own encryption keys, provide all the following headers in the request.

**Note**

If you use this feature, the `ETag` value that Amazon S3 returns in the response is not the MD5 of the object.

Name	Description	Required
<code>x-amz-server-side-encryption-customer-algorithm</code>	<p>Specifies the algorithm to use to when encrypting the object. Type: String Default: None Valid Value: <code>AES256</code> Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
<code>x-amz-server-side-encryption-customer-key</code>	<p>Specifies the customer-provided base64-encoded encryption key that Amazon S3 should use to encrypt data. Amazon S3 uses this value to store the object and then discards it. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the <code>x-amz-server-side-encryption-customer-algorithm</code> header. Type: String Default: None Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
<code>x-amz-server-side-encryption-customer-key-MD5</code>	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.</p>	Yes

Name	Description	Required
	<p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</p>	

For more information on Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C), see [Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys \(SSE-C\)](#) in the [Amazon Simple Storage Service Developer Guide](#).

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
<code>x-amz-expiration</code>	If the expiration is configured for the object (see <a href="#">PUT Bucket lifecycle (p. 265)</a> ), the response includes this header. It includes the <code>expiry-date</code> and <code>rule-id</code> key-value pairs that provide information about object expiration. The value of the <code>rule-id</code> is URL encoded.  Type: String
<code>x-amz-server-side-encryption</code>	If you specified server-side encryption either with an AWS KMS-managed or Amazon S3-managed encryption key in your PUT request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.  Type: String
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code> , this header specifies the ID of the AWS KMS master encryption key that was used for the object.  Type: String
<code>x-amz-server-side-encryption-customer-algorithm</code>	If server-side encryption with customer-provided encryption keys encryption was requested, the response includes this header that confirms the encryption algorithm that was used.  Type: String  Valid Values: AES256
<code>x-amz-server-side-encryption-customer-key-md5</code>	If server-side encryption using customer-provided encryption keys was requested, the response returns this header to verify the roundtrip message integrity of the customer-provided encryption key.

Name	Description
customer-key-MD5	Type: String
x-amz-version-id	Version of the object. Type: String

## Response Elements

This implementation of the operation does not return response elements.

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Upload an Object

#### Sample Request

The following request stores the my-image.jpg image in the myBucket bucket.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-meta-author: Janet
Expect: 100-continue
[11434 bytes of object data]
```

#### Sample Response with Versioning Suspended

```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

If an expiration rule that was created on the bucket using lifecycle configuration applies to the object, you get a response with an x-amz-expiration header as shown in the following response. For more information, see [Transitioning Objects: General Considerations](#) in the *Amazon Simple Storage Service Developer Guide*.

```
HTTP/1.1 100 Continue
```

```
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiiIgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-expiration: expiry-date="Fri, 23 Dec 2012 00:00:00 GMT", rule-id="1"
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Sample Response with Versioning Enabled

If the bucket has versioning enabled, the response includes the `x-amz-version-id` header.

```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiiIgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fjj3HHNVfdsQUIFDnsidf038jfdsjGFDSIRp
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fbacf535f27731c9771645a39863328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Example 2: Upload an Object (Specify Storage Class)

### Sample Request: Specifying the Reduced Redundancy Storage Class

The following request stores the image, `my-image.jpg`, in the `myBucket` bucket. The request specifies the `x-amz-storage-class` header to request that the object is stored using the `REDUCED_REDUNDANCY` storage class.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: REDUCED_REDUNDANCY
```

### Sample Response

```
HTTP/1.1 100 Continue

HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiiIgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Example 3:Upload an Object (Specify Access Permission Explicitly)

### Sample Request: Uploading an Object and Specifying Access Permissions Explicitly

The following request stores the `TestObject.txt` file in the `myBucket` bucket. The request specifies various ACL headers to grant permission to AWS accounts that are specified with a canonical user ID and an email address.

```
PUT TestObject.txt HTTP/1.1
Host: myBucket.s3.amazonaws.com
x-amz-date: Fri, 13 Apr 2012 05:40:14 GMT
Authorization: authorization string
x-amz-grant-write-acp: id=8a6925ce4adf588a4532142d3f74dd8c71fa124ExampleCanonicalUserID
x-amz-grant-full-control: emailAddress="ExampleUser@amazon.com"
x-amz-grant-write: emailAddress="ExampleUser1@amazon.com",
    emailAddress="ExampleUser2@amazon.com"
Content-Length: 300
Expect: 100-continue
Connection: Keep-Alive

...Object data in the body...
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: RUxG2sZJUFs+ezeAS2i0Xj6w/ST6xqF/8pFNHjTjTrECW56SCAUWGg+7QLVoj1GH
x-amz-request-id: 8D017A90827290BA
Date: Fri, 13 Apr 2012 05:40:25 GMT
ETag: "dd038b344cf9553547f8b395a814b274"
Content-Length: 0
Server: AmazonS3
```

## Example 4: Upload an Object (Specify Access Permission Using Canned ACL)

### Sample Request: Using a Canned ACL to Set Access Permissions

The following request stores the `TestObject.txt` file in the `myBucket` bucket. The request uses an `x-amz-acl` header to specify a canned ACL that grants READ permission to the public.

```
...Object data in the body...
PUT TestObject.txt HTTP/1.1
Host: myBucket.s3.amazonaws.com
x-amz-date: Fri, 13 Apr 2012 05:54:57 GMT
x-amz-acl: public-read
Authorization: authorization string
Content-Length: 300
Expect: 100-continue
Connection: Keep-Alive

...Object data in the body...
```

### Sample Response

```
HTTP/1.1 200 OK
```

```
x-amz-id-2: Yd6PSJxJFQeTYJ/3dDO7miqJfVMXXW0S2Hijo3WFs4bz6oe2QCVXasxXLZdMfASd
x-amz-request-id: 80DF413BB3D28A25
Date: Fri, 13 Apr 2012 05:54:59 GMT
ETag: "dd038b344cf9553547f8b395a814b274"
Content-Length: 0
Server: AmazonS3
```

## Example 5: Upload an Object (Request Server-Side Encryption Using a Customer-Provided Encryption Key)

This example of an upload object requests server-side encryption and provides an encryption key.

```
PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: /*
Authorization:authorization string
Date: Wed, 28 May 2014 19:31:11 +0000
x-amz-server-side-encryption-customer-key:g01CfA3Dv40jZz5SQJ1ZukLRFqtI5WorC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5:ZjQrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm:AES256
```

In the response, Amazon S3 returns the encryption algorithm and MD5 of the encryption key that you specified when uploading the object. The ETag that is returned is not the MD5 of the object.

```
HTTP/1.1 200 OK
x-amz-id-2: 7qoYGN7uMuFuYS6m7a4lszH6in+hccE+4DXPmDZ7C9KqucjnZC1gI5mshai6fbMG
x-amz-request-id: 06437EDD40C407C7
Date: Wed, 28 May 2014 19:31:12 GMT
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjQrne1X/iTcskbY2example
ETag: "ae89237c20e759c5f479ece02c642f59"
```

## Example 6: Upload an Object and Specify Tags

This example of an upload object request specifies the optional `x-amz-tagging` header to add tags to the object.

```
PUT /example-object HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Accept: /*
Authorization:authorization string
Date: Thu, 22 Sep 2016 21:58:13 GMT
x-amz-tagging: tag1=value1&tag2=value2
[... bytes of object data]
```

After the object is created, Amazon S3 stores the specified object tags in the tagging subresource that is associated with the object.

```
HTTP/1.1 200 OK
x-amz-id-2: 7qoYGN7uMuFuYS6m7a4lszH6in+hccE+4DXPmDZ7C9KqucjnZC1gI5mshai6fbMG
x-amz-request-id: 06437EDD40C407C7
Date: Thu, 22 Sep 2016 21:58:17 GMT
```

## Related Resources

- [PUT Object - Copy \(p. 431\)](#)

- [POST Object \(p. 385\)](#)
- [GET Object \(p. 349\)](#)

# PUT Object legal hold

Service: Amazon Simple Storage Service

Applies a Legal Hold configuration to the specified object.

## Request Syntax

```
PUT /<object-key>?legal-hold&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests \(AWS Signature Version 4\))
```

## URI Request Parameters

### versionId

The version ID of the object version that you want to put a retention period on.

## Request Body

The request accepts the following data in XML format.

### [LegalHold \(p. 427\)](#)

Root level tag for the LegalHold parameters.

Required: Yes

### [Status \(p. 427\)](#)

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

Required: Yes

Example Request Body:

```
<LegalHold>
  <Status>ON</Status>
</LegalHold>
```

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

## Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

# PUT Object retention

Service: Amazon Simple Storage Service

Places an Object Retention configuration on an object.

## Request Syntax

```
PUT /<object-key>?retention&versionId=<version-id> HTTP/1.1
Host: <bucket-name>.s3.amazonaws.com
Date: <Thu, 15 Nov 2016 00:17:21 GMT>
Authorization: <authorization-string> (see Authenticating Requests \(AWS Signature Version 4\))
```

## URI Request Parameters

### versionId

The version ID of the object version that you want to put a retention period on.

## Request Body

The request accepts the following data in XML format.

### [Retention \(p. 429\)](#)

Root level tag for the Retention parameters.

Required: Yes

### [Mode \(p. 429\)](#)

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: No

#### Note

If either Retention or Mode are specified, then both must be present.

### [RetainUntilDate \(p. 429\)](#)

The date and time when the retention period expires.

Type: Timestamp

Format: `2020-01-05T00:00:00.000Z`

Required: No

Example Request Body:

```
<Retention>
  <Mode>GOVERNANCE</Mode>
```

```
<RetainUntilDate>2020-01-05T00:00:00.000Z</RetainUntilDate>
</Retention>
```

## Response Syntax

```
HTTP/1.1 200
```

## Response Elements

If the action is successful, the service sends back an HTTP 200 response.

## Related Resources

[Locking Objects](#) in the *Amazon Simple Storage Service Developer Guide*.

# PUT Object - Copy

## Description

This implementation of the `PUT` operation creates a copy of an object that is already stored in Amazon S3. A `PUT` copy operation is the same as performing a `GET` and then a `PUT`. Adding the request header, `x-amz-copy-source`, makes the `PUT` operation copy the source object into the destination bucket.

### Note

You can store individual objects of up to 5 TB in Amazon S3. You create a copy of your object up to 5 GB in size in a single atomic operation using this API. However, for copying an object greater than 5 GB, you must use the multipart upload [Upload Part - Copy \(p. 514\)](#) API. For conceptual information, see [Copy Object Using the REST Multipart Upload API](#) in the *Amazon Simple Storage Service Developer Guide*.

When copying an object, you can preserve most of the metadata (default) or specify new metadata. However, the ACL is not preserved and is set to `private` for the user making the request.

### Important

Amazon S3 Transfer Acceleration does not support cross-region copies. If you request a cross-region copy using a Transfer Acceleration endpoint, you get a `400 Bad Request` error. For more information about transfer acceleration, see [Transfer Acceleration](#) in the *Amazon Simple Storage Service Developer Guide*.

All copy requests must be authenticated and cannot contain a message body. Additionally, you must have `READ` access to the source object and `WRITE` access to the destination bucket. For more information, see [REST Authentication](#).

To copy an object only under certain conditions, such as whether the `ETag` matches or whether the object was modified before or after a specified date, use the request headers `x-amz-copy-source-if-match`, `x-amz-copy-source-if-none-match`, `x-amz-copy-source-if-unmodified-since`, or `x-amz-copy-source-if-modified-since`.

### Note

All headers with the `x-amz-` prefix, including `x-amz-copy-source`, must be signed.

You can use this operation to change the storage class of an object that is already stored in Amazon S3 using the `x-amz-storage-class` request header. For more information, go to [Storage Classes](#) in the *Amazon Simple Storage Service Developer Guide*.

The source object that you are copying can be encrypted or unencrypted. If the source object is encrypted, it can be encrypted by server-side encryption using AWS-managed encryption keys or by using a customer-provided encryption key. When copying an object, you can request that Amazon S3 encrypt the target object by using either the AWS-managed encryption keys or by using your own encryption key. You can do this regardless of the form of server-side encryption that was used to encrypt the source, or even if the source object was not encrypted. For more information about server-side encryption, see [Using Server-Side Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

A copy request might return an error when Amazon S3 receives the copy request or while Amazon S3 is copying the files. If the error occurs before the copy operation starts, you receive a standard Amazon S3 error. If the error occurs during the copy operation, the error response is embedded in the `200 OK` response. This means that a `200 OK` response can contain either a success or an error. Design your application to parse the contents of the response and handle it appropriately.

If the copy is successful, you receive a response with information about the copied object.

### Note

If the request is an HTTP 1.1 request, the response is chunk encoded. If it were not, it would not contain the `content-length`, and you would need to read the entire body.

The copy request charge is based on the storage class and region you specify for the destination object. For pricing information, see [Amazon S3 Pricing](#).

## Versioning

By default, `x-amz-copy-source` identifies the current version of an object to copy. (If the current version is a delete marker, Amazon S3 behaves as if the object was deleted.) To copy a different version, use the `versionId` subresource.

If you enable versioning on the target bucket, Amazon S3 generates a unique version ID for the object being copied. This version ID is different from the version ID of the source object. Amazon S3 returns the version ID of the copied object in the `x-amz-version-id` response header in the response.

If you do not enable versioning or suspend it on the target bucket, the version ID that Amazon S3 generates is always `null`.

If the source object's storage class is `GLACIER`, then you must restore a copy of this object before you can use it as a source object for the copy operation. For more information, see [POST Object restore \(p. 397\)](#).

To see sample requests that use versioning, see [Sample Request: Copying a specified version of an object \(p. 444\)](#).

## Access Permissions

When copying an object, you can optionally specify the accounts or groups that should be granted specific permissions on the new object. There are two ways to grant the permissions using the request headers:

- Specify a canned ACL with the `x-amz-acl` request header. For more information, see [Canned ACL](#) in the *Amazon Simple Storage Service Developer Guide*.
- Specify access permissions explicitly with the `x-amz-grant-read`, `x-amz-grant-read-acp`, `x-amz-grant-write-acp`, and `x-amz-grant-full-control` headers. These headers map to the set of permissions that Amazon S3 supports in an ACL. For more information, go to [Access Control List \(ACL\) Overview](#) in the *Amazon Simple Storage Service Developer Guide*.

### Note

You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

## Requests

### Syntax

```
PUT /destinationObject HTTP/1.1
Host: destinationBucket.s3.amazonaws.com
x-amz-copy-source: /source_bucket/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
<request metadata>
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Date: date
```

**Note**

The syntax shows only some of the request headers. For a complete list, see the Request Headers section.

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
x-amz-copy-source	<p>The name of the source bucket and key name of the source object, separated by a slash (/).</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints:</p> <p>This string must be URL-encoded. Additionally, the source bucket must be valid and you must have READ access to the valid source object.</p> <p>If the source object is archived in Amazon S3 Glacier (the storage class of the object is GLACIER), you must first restore a temporary copy using the <a href="#">POST Object restore (p. 397)</a>. Otherwise, Amazon S3 returns the 403 <code>ObjectNotInActiveTierError</code> error response.</p>	Yes
x-amz-metadata-directive	<p>Specifies whether the metadata is copied from the source object or is replaced with metadata provided in the request.</p> <ul style="list-style-type: none"> <li>If the metadata is copied, all of the metadata except for the version ID remains unchanged. In addition, the <code>server-side-encryption</code>, <code>storage-class</code> and <code>website-redirect-location</code> metadata from the source is not copied. If you specify this metadata explicitly in the copy request, Amazon S3 adds this metadata to the resulting object. If you specify headers in the request that specifies user-defined metadata, Amazon S3 ignores these headers.</li> <li>If the metadata is replaced, all of the original metadata is replaced by the metadata that you specify.</li> </ul> <p>Type: String</p>	No

Name	Description	Required
	<p>Default: COPY</p> <p>Valid values: COPY   REPLACE</p> <p>Constraints: Values other than COPY or REPLACE result in an immediate 400-based error response. You can't copy an object to itself unless you specify the <code>MetadataDirective</code> header and set its value to REPLACE.</p> <p>For information on supported metadata, see <a href="#">Common Request Headers (p. 2)</a></p>	
<code>x-amz-copy-source-if-match</code>	<p>Copies the object if its entity tag (<code>ETag</code>) matches the specified tag. Otherwise, the request returns a 412 HTTP status code error (failed precondition).</p> <p>For more information, see Consideration 1 after this table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: This header can be used with <code>x-amz-copy-source-if-unmodified-since</code>, but it cannot be used with other conditional copy headers.</p>	No
<code>x-amz-copy-source-if-none-match</code>	<p>Copies the object if its entity tag (<code>ETag</code>) is different than the specified <code>ETag</code>. Otherwise, the request returns a 412 HTTP status code error (failed precondition).</p> <p>For more information, see Consideration 1 after this table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: This header can be used with <code>x-amz-copy-source-if-modified-since</code>, but it cannot be used with other conditional copy headers.</p>	No

Name	Description	Required
x-amz-copy-source-if-unmodified-since	<p>Copies the object if it hasn't been modified since the specified time. Otherwise, the request returns a 412 HTTP status code error (failed precondition).</p> <p>For more information, see Consideration 1 after this table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: This must be a valid HTTP date. This header can be used with x-amz-copy-source-if-match, but cannot be used with other conditional copy headers.</p>	No
x-amz-copy-source-if-modified-since	<p>Copies the object if it has been modified since the specified time; otherwise, the request returns a 412 HTTP status code error (failed condition).</p> <p>For more information, see Consideration 2 after this table.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: This must be a valid HTTP date. This header can be used with x-amz-copy-source-if-none-match, but cannot be used with other conditional copy headers.</p>	No
x-amz-storage-class	<p>If you don't specify this header, Amazon S3 uses STANDARD, the default, for the storage class. Amazon S3 supports other storage classes. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Enum</p> <p>Default: STANDARD</p> <p>Valid Values: STANDARD   STANDARD_IA   ONEZONE_IA   INTELLIGENT_TIERING   GLACIER   REDUCED_REDUNDANCY</p>	No

Name	Description	Required
x-amz-tagging-directive	<p>Specifies whether the object tags are copied from the source object or replaced with tags provided in the request.</p> <ul style="list-style-type: none"> <li>If the tags are copied, the tagset remains unchanged.</li> <li>If the tags are replaced, all of the original tagset is replaced by the tags you specify.</li> </ul> <p>If you don't specify a tagging directive, Amazon S3 copies tags by default.</p> <p>If the tagging directive is REPLACE, you specify any tags in url format in the x-amz-tagging header, similar to using a PUT object with tags.</p> <p>If the tagging directive is REPLACE, but you don't specify the x-amz-tagging in the request, the destination object won't have tags.</p> <p>Type: String</p> <p>Default: COPY</p> <p>Valid values: COPY   REPLACE</p> <p>Constraints: Values other than COPY or REPLACE result in an immediate 400-based error response.</p>	No

Name	Description	Required
x-amz-website-redirect-location	<p>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see <a href="#">Object Key and Metadata</a>.</p> <p>In the following example, the request header sets the redirect to an object (anotherPage.html) in the same bucket:</p> <pre>x-amz-website-redirect-location: /anotherPage.html</pre> <p>In the following example, the request header sets the object redirect to another website:</p> <pre>x-amz-website-redirect-location: http://www.example.com/</pre> <p>For more information about website hosting in Amazon S3, see <a href="#">Hosting Websites on Amazon S3</a> and <a href="#">How to Configure Website Page Redirects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The value must be prefixed by, "/", "http://" or "https://". The length of the value is limited to 2 K.</p>	No

Consider the following when using request headers:

- **Consideration 1** – If both the x-amz-copy-source-if-match and x-amz-copy-source-if-unmodified-since headers are present in the request and evaluate as follows, Amazon S3 returns 200 OK and copies the data:

x-amz-copy-source-if-match condition evaluates to true

x-amz-copy-source-if-unmodified-since condition evaluates to false

- **Consideration 2** – If both of the x-amz-copy-source-if-none-match and x-amz-copy-source-if-modified-since headers are present in the request and evaluate as follows, Amazon S3 returns the 412 Precondition Failed response code:

x-amz-copy-source-if-none-match condition evaluates to false

x-amz-copy-source-if-modified-since condition evaluates to true

## Server-Side- Encryption-Specific Request Headers

To encrypt the target object, you must provide the appropriate encryption-related request headers. The one you use depends on whether you want to use AWS-managed encryption keys or provide your own encryption key:

- To encrypt the target object using server-side encryption with an AWS-managed encryption key, provide the following request headers, as appropriate.

Name	Description	Required
x-amz-server-side-encryption	<p>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</p> <p>Type: String</p> <p>Valid Value: aws:kms, AES256</p>	Yes
x-amz-server-side-encryption-aws-kms-key-id	<p>If the x-amz-server-side-encryption header is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (AWS KMS) master encryption key that was used for the object.</p> <p>Type: String</p>	Yes, if the value of x-amz-server-side-encryption is aws:kms
x-amz-server-side-encryption-context	<p>If x-amz-server-side-encryption is present and its value is aws:kms, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.</p> <p>Type: String</p>	No

### Note

If you specify x-amz-server-side-encryption:aws:kms, but don't provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

### Important

All GET and PUT requests for an object protected by AWS KMS fail if you don't make them with SSL or by using SigV4.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see [Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys](#) in the *Amazon Simple Storage Service Developer Guide*.

- To encrypt the target object using server-side encryption with an encryption key that you provide, use the following headers.

Name	Description	Required
x-amz-server-side-encryption-customer-algorithm	<p>Specifies the algorithm to use when encrypting the object.</p> <p>Type: String</p>	Yes

Name	Description	Required
	<p>Default: None</p> <p>Valid Value: AES256</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	
<code>x-amz-server-side-encryption-customer-key</code>	<p>Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use to encrypt data. Amazon S3 uses this value to store the object and then discards it. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the <code>x-amz-server-side-encryption-customer-algorithm</code> header.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
<code>x-amz-server-side-encryption-customer-key-MD5</code>	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header as a message integrity check to ensure that the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</p>	Yes

- If the source object is encrypted using server-side encryption with customer-provided encryption keys, you must use the following headers.

Name	Description	Required
<code>x-amz-copy-source-server-side-encryption-customer-algorithm</code>	<p>Specifies the algorithm to use when decrypting the source object.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Value: AES256</p> <p>Constraints: Must be accompanied by valid <code>x-amz-copy-source-server-side-encryption-customer-key</code> and <code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
<code>x-amz-copy-source-server-side</code>	<p>Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use to decrypt the source object. After the copy operation, Amazon S3 discards this key. The</p>	Yes

Name	Description	Required
-encryption-customer-key	<p>encryption key provided in this header must be one that was used when the source object was created.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-copy-source-server-side-encryption-customer-algorithm</code> and <code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code> headers.</p>	
<code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code>	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-copy-source-server-side-encryption-customer-algorithm</code> and <code>x-amz-copy-source-server-side-encryption-customer-key</code> headers.</p>	Yes

For more information on Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C), see [Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys \(SSE-C\)](#) in the [Amazon Simple Storage Service Developer Guide](#).

## Access-Control-List-ACL)-Specific Request Headers

You also can use the following access control-related headers with this operation. By default, all objects are private. Only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or to predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see [Using ACLs](#). With this operation, you can grant access permissions using one of the following two methods:

- **Specify a canned ACL** — Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see [Canned ACL](#).

Name	Description	Required
<code>x-amz-acl</code>	<p>The canned ACL to apply to the object.</p> <p>Type: String</p> <p>Default: <code>private</code></p> <p>Valid Values: <code>private</code>   <code>public-read</code>   <code>public-read-write</code>   <code>aws-exec-read</code>   <code>authenticated-read</code>   <code>bucket-owner-read</code>   <code>bucket-owner-full-control</code></p> <p>Constraints: None</p>	No

- **Specify access permissions explicitly** — To explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each header maps to specific permissions that Amazon S3 supports in an ACL. For more information, see [Access Control List \(ACL\) Overview](#). In the header, you specify a list of grantees who get the specific permission.

Name	Description	Required
x-amz-grant-read	<p>Gives the grantee permissions to read the object data and its metadata.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-write	<p>Not applicable. This header applies only when granting access permissions on a bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-read-acp	<p>Gives the grantee permissions to read the object ACL.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-write-acp	<p>Gives the grantee permissions to write the ACL for the applicable object.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-full-control	<p>Gives the grantee READ, READ_ACP, and WRITE_ACP permissions on the object.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No

You specify each grantee as a `type=value` pair, where the type is one of the following:

- **emailAddress** – if the value specified is the email address of an AWS account
- **id** – if the value specified is the canonical user ID of an AWS account
- **uri** – if you are granting permissions to a predefined group.

For example, the following `x-amz-grant-read` header grants the AWS accounts identified by email addresses permissions to read object data and its metadata:

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
<code>x-amz-expiration</code>	If an <code>Expiration</code> action is configured for the object as part of the bucket's lifecycle configuration, Amazon S3 returns this header. The header value includes an "expiry-date" component and a URL-encoded "rule-id" component. For version-enabled buckets, this header applies only to current versions. Amazon S3 does not provide a header to infer when a noncurrent version is eligible for permanent deletion. For more information, see <a href="#">PUT Bucket lifecycle (p. 265)</a> .  Type: String
<code>x-amz-copy-source-version-id</code>	Version of the source object that was copied.  Type: String
<code>x-amz-server-side-encryption</code>	If you specified server-side encryption either with an encryption key managed by AWS KMS or Amazon S3 in your copy request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.  Type: String
<code>x-amz-server-side-encryption-aws-kms-key-id</code>	If the <code>x-amz-server-side-encryption</code> header is present and has the value of <code>aws:kms</code> , this header specifies the ID of the AWS KMS master encryption key that was used for the object.  Type: String
<code>x-amz-server-side-encryption-customer-algorithm</code>	If server-side encryption with customer-provided encryption keys (SSE-C) encryption was requested, the response includes this header, which confirms the encryption algorithm used for the destination object.  Type: String  Valid values: <code>AES256</code>

Name	Description
x-amz-server-side-encryption-customer-key-MD5	If SSE-C encryption was requested, the response includes this header to verify the integrity of the roundtrip message of the customer-provided encryption key that was used to encrypt the destination object.  Type: String
x-amz-storage-class	Provides information about the object's storage class. Amazon S3 returns this header for all objects except Standard storage class objects.  For more information, see <a href="#">Storage Classes</a> in <i>Amazon Simple Storage Service Developer Guide</i> .  Type: String  Default: None
x-amz-version-id	Version of the copied object in the destination bucket.  Type: String

## Response Elements

Name	Description
CopyObjectResult	Container for all response elements.  Type: Container  Ancestor: None
ETag	Returns the ETag of the new object. The ETag reflects only changes to the contents of an object, not its metadata. The source and destination ETag is identical for a successfully copied object.  Type: String  Ancestor: CopyObjectResult
LastModified	Returns the date that the object was last modified.  Type: String  Ancestor: CopyObjectResult

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This example copies `my-image.jpg` into the bucket `bucket`, with the key name `my-second-image.jpg`.

```
PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
x-amz-copy-source: /bucket/my-image.jpg
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-copy-source-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
+MTRCx3vjbVH40Nr8X8gdRQBpUMLUo
x-amz-version-id: QUpfdndhfd8438MNFDN93jdnJFkdmqn893
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

`x-amz-version-id` returns the version ID of the object in the destination bucket. `x-amz-copy-source-version-id` returns the version ID of the source object.

### Sample Request: Copying a Specified Version of an Object

The following request copies the `my-image.jpg` key with the specified version ID, copies it into the bucket `bucket`, and gives it the `my-second-image.jpg` key.

```
PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
x-amz-copy-source: /bucket/my-image.jpg?versionId=3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
+MTRCx3vjbVH40Nr8X8gdRQBpUMLUo
Authorization: authorization string
```

### Success Response: Copying a Versioned Object into a Version-enabled Bucket

The following response shows that an object was copied into a target bucket where versioning is enabled.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
```

```
x-amz-version-id: QUpfdndhfd8438MNFDN93jdnJFkdmqnh893
x-amz-copy-source-version-id: 09df8234529fjs0dfi0w52935029wefdj
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
    <LastModified>2009-10-28T22:32:00</LastModified>
    <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

## Success Response: Copying a Versioned Object into a Version-suspended Bucket

The following response shows that an object was copied into a target bucket where versioning is suspended. The parameter <VersionId> does not appear.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-copy-source-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
+MTRCx3VjVBH4ONr8X8gdRQBpUMLUo
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
    <LastModified>2009-10-28T22:32:00</LastModified>
    <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

## Sample: Copy from Unencrypted Object to an Object Encrypted with Server-side Encryption with Customer-provided Encryption Keys

The following example specifies the HTTP PUT header to copy an unencrypted object to an object encrypted with server-side encryption with customer-provided encryption keys (SSE-C).

```
PUT /exampleDestinationObject HTTP/1.1
Host: example-destination-bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key: Base64(YourKey)
x-amz-server-side-encryption-customer-key-MD5 : Base64(MD5(YourKey))
x-amz-metadata-directive: metadata_directive
x-amz-copy-source: /example_source_bucket/exampleSourceObject
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
<request metadata>
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Date: date
```

## Sample: Copy from an Object Encrypted with SSE-C to an Object Encrypted with SSE-C

The following example specifies the HTTP `PUT` header to copy an object encrypted with server-side encryption with customer-provided encryption keys to an object encrypted with server-side encryption with customer-provided encryption keys for key rotation.

```
PUT /exampleDestinationObject HTTP/1.1
Host: example-destination-bucket.s3.amazonaws.com
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key: Base64(NewKey)
x-amz-server-side-encryption-customer-key-MD5: Base64(MD5(NewKey))
x-amz-metadata-directive: metadata_directive
x-amz-copy-source: /source_bucket/sourceObject
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
x-amz-copy-source-server-side-encryption-customer-algorithm: AES256
x-amz-copy-source-server-side-encryption-customer-key: Base64(OldKey)
x-amz-copy-source-server-side-encryption-customer-key-MD5: Base64(MD5(OldKey))
<request metadata>
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
Date: date
```

## Related Resources

- [Copying Objects](#)
- [PUT Object \(p. 412\)](#)
- [GET Object \(p. 349\)](#)

# PUT Object acl

## Description

This implementation of the `PUT` operation uses the `acl` subresource to set the access control list (ACL) permissions for an object that already exists in a bucket. You must have `WRITE_ACP` permission to set the ACL of an object.

You can use one of the following two ways to set an object's permissions:

- Specify the ACL in the request body, or
- Specify permissions using request headers

Depending on your application needs, you may choose to set the ACL on an object using either the request body or the headers. For example, if you have an existing application that updates an object ACL using the request body, then you can continue to use that approach.

## Versioning

The ACL of an object is set at the object version level. By default, `PUT` sets the ACL of the current version of an object. To set the ACL of a different version, use the `versionId` subresource.

To see sample requests that use versioning, see [Sample Request: Setting the ACL of a specified object version \(p. 452\)](#).

## Requests

### Syntax

The following request shows the syntax for sending the ACL in the request body. If you want to use headers to specify the permissions for the object, you cannot send the ACL in the request body. Instead, see the Request Headers section for a list of headers you can use.

```
PUT /ObjectName?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))  
  
<AccessControlPolicy>
  <Owner>
    <ID>ID</ID>
    <DisplayName>EmailAddress</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="CanonicalUser">
        <ID>ID</ID>
        <DisplayName>EmailAddress</DisplayName>
      </Grantee>
      <Permission>Permission</Permission>
    </Grant>
    ...
  </AccessControlList>
</AccessControlPolicy>
```

**Note**

The syntax shows some of the request headers. For a complete list see the Request Headers section.

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

You can use the following request headers in addition to the [Common Request Headers \(p. 2\)](#).

### Access Control List (ACL) Specific Request Headers

These headers enable you to set access permissions using one of the following methods:

- Specify canned ACL, or
- Specify the permission for each grantee explicitly

Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined a set of grantees and permissions. For more information, see [Canned ACL](#). To grant access permissions by specifying canned ACLs, you use the following header and specify the canned ACL name as its value. If you use this header, you cannot use other access control-specific headers in your request.

Name	Description	Required
x-amz-acl	<p>Sets the ACL of the object using the specified canned ACL. For more information, go to <a href="#">Canned ACL</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Valid Values: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</p> <p>Default: private</p>	No

If you need to grant individualized access permissions on an object, you can use the following x-amz-grant-permission headers. When using these headers you specify explicit access permissions and grantees (AWS accounts or Amazon S3 groups) who will receive the permission. If you use these ACL specific headers, you cannot use x-amz-acl header to set a canned ACL.

**Note**

Each of the following request headers maps to specific permissions Amazon S3 supports in an ACL. For more information, go to [Access Control List \(ACL\) Overview](#).

Name	Description	Required
x-amz-grant-read	<p>Allows the specified grantee to list the objects in the bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No

Name	Description	Required
x-amz-grant-write	<p>Not applicable when granting access permissions on objects. You can use this when granting access permissions on buckets.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-read-acp	<p>Allows the specified grantee to read the bucket ACL.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-write-acp	<p>Allows the specified grantee to write the ACL for the applicable bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-full-control	<p>Allows the specified grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No

For each of these headers, the value is a comma-separated list of one or more grantees. You specify each grantee as a `type=value` pair, where the type can be one of the following:

- **emailAddress** — if value specified is the email address of an AWS account
- **id** — if value specified is the canonical user ID of an AWS account
- **uri** — if granting permission to a predefined group.

For example, the following `x-amz-grant-read` header grants list objects permission to the two AWS accounts identified by their email addresses.

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

For more information, go to [Access Control List \(ACL\) Overview](#).

## Request Elements

If you decide to use the request body to specify an ACL, you must use the following elements.

**Note**

If you use the request body, you cannot use the request headers to set an ACL.

Name	Description	Required
AccessControlList	Container for ACL information  Type: Container  Ancestors: AccessControlPolicy	No
AccessControlPolicy	Contains the elements that set the ACL permissions for an object per grantee  Type: Container  Ancestors: None	No
DisplayName	Screen name of the bucket owner  Type: String  Ancestors: AccessControlPolicy.Owner	No
Grant	Container for the grantee and his or her permissions  Type: Container  Ancestors: AccessControlPolicy.AccessControlList	No
Grantee	The subject whose permissions are being set.  Type: String  Valid Values: DisplayName   EmailAddress   AuthenticatedUser. For more information, see <a href="#">Grantee Values (p. 451)</a> .  Ancestors: AccessControlPolicy.AccessControlList.Grant	No
ID	ID of the bucket owner, or the ID of the grantee  Type: String  Ancestors: AccessControlPolicy.Owner or AccessControlPolicy.AccessControlList.Grant	No
Owner	Container for the bucket owner's display name and ID  Type: Container  Ancestors: AccessControlPolicy	Yes
Permission	Specifies the permission given to the grantee  Type: String  Valid Values: FULL_CONTROL   WRITE   WRITE_ACP   READ   READ_ACP  Ancestors: AccessControlPolicy.AccessControlList.Grant	No

## Grantee Values

You can specify the person (grantee) to whom you're assigning access rights (using request elements) in the following ways:

- By the person's ID:

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

DisplayName is optional and ignored in the request.

- By Email address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="AmazonCustomerByEmail"><EmailAddress><replaceable>Grantees@email.com</
replaceable></EmailAddress>lt;/Grantee>
```

The grantee is resolved to the CanonicalUser and, in a response to a GET Object acl request, appears as the CanonicalUser.

- By URI:

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

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
x-amz-version-id	Version of the object whose ACL is being set.  Type: String  Default: None

### Response Elements

This operation does not return response elements.

### Special Errors

This operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request grants access permission to an existing object. The request specifies the ACL in the body. In addition to granting full control to the object owner, the XML specifies full control to an AWS account identified by its canonical user ID.

```
PUT /my-image.jpg?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
        <DisplayName>CustomerName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

### Sample Response

The following shows a sample response when versioning on the bucket is enabled.

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51T9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 3/L4kqtJlcpXrof3vjVBH40Nr8X8gdRQBpUMLUo
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

### Sample Request: Setting the ACL of a specified object version

The following request sets the ACL on the specified version of the object.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCx3vjbh40Nrjfkd
HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: authorization string
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
```

```
<DisplayName>mtd@amazon.com</DisplayName>
</Owner>
<AccessControlList>
  <Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="CanonicalUser">
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51u8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 3/L4kqtJlcpXro3vjVBH40Nr8X8gdRQBpUMLUo
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Sample Request: Access permissions specified using headers

The following request uses ACL-specific request headers, `x-amz-acl`, and specifies a canned ACL (`public_read`) to grant object read access to everyone.

```
PUT ExampleObject.txt?acl HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-acl: public-read
Accept: */
Authorization: authorization string
Host: s3.amazonaws.com
Connection: Keep-Alive
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: w5YegkbG6ZDsje4WK56RWPxNQHIQ0CjrjyRVFZhEJI9E3kbabXnB09w5G7Dmxsgk
x-amz-request-id: C13B2827BD8455B1
Date: Sun, 29 Apr 2012 23:24:12 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [PUT Object - Copy \(p. 431\)](#)
- [POST Object \(p. 385\)](#)
- [GET Object \(p. 349\)](#)

# PUT Object tagging

## Description

This implementation of the `PUT` operation uses the `tagging` subresource to add a set of tags to an existing object.

A tag is a key-value pair. You can associate tags with an object by sending a `PUT` request against the `tagging` subresource that is associated with the object. You can retrieve tags by sending a `GET` request. For more information, see [GET Object tagging \(p. 368\)](#).

For tagging-related restrictions related to characters and encodings, see [Tag Restrictions](#) in the *AWS Billing and Cost Management User Guide*. Note that Amazon S3 limits the maximum number of tags to 10 tags per object.

To use this operation, you must have permission to perform the `s3:PutObjectTagging` action. By default, the bucket owner has this permission and can grant this permission to others.

To put tags of any other version, use the `versionId` query parameter. You also need permission for the `s3:PutObjectVersionTagging` action.

For information about the Amazon S3 object tagging feature, see [Object Tagging](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

The following request shows the syntax for sending tagging information in the request body.

```
PUT /ObjectName?tagging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
<Tagging>
  <TagSet>
    <Tag>
      <Key>Tag Name</Key>
      <Value>Tag Value</Value>
    </Tag>
  </TagSet>
</Tagging>
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

`Content-MD5` is a required header for this operation.

### Request Elements

Name	Description	Required
<code>Tagging</code>	Container for the <code>TagSet</code> and <code>Tag</code> elements.	Yes

Name	Description	Required
	Type: String  Ancestors: None	
TagSet	Container for a set of tags  Type: Container  Ancestors: Tagging	Yes
Tag	Container for tag information.  Type: Container  Ancestors: TagSet	No
Key	Name of the tag.  Type: String  Ancestors: Tag	Yes, if Tag is specified.
Value	Value of the tag.  Type: String  Ancestors: Tag	Yes, if Tag is specified.

## Responses

### Response Headers

The operation returns response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not return response elements.

### Special Errors

- InvalidTagError - The tag provided was not a valid tag. This error can occur if the tag did not pass input validation. For more information, see [Object Tagging](#) in the *Amazon Simple Storage Service Developer Guide*.
- MalformedXMLError - The XML provided does not match the schema.
- OperationAbortedError - A conflicting conditional operation is currently in progress against this resource. Please try again.
- InternalError - The service was unable to apply the provided tag to the object.

## Examples

### Sample Request: Add tag set to an object

The following request adds a tag set to the existing object `object-key` in the `examplebucket` bucket.

```
PUT object-key?tagging HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Content-Length: length
Content-MD5: pUNXr/BjKK5G2UKEexample==
x-amz-date: 20160923T001956Z
Authorization: authorization string
<Tagging>
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
</Tagging>
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Fri, 23 Sep 2016 00:20:19 GMT
```

## Related Resources

- [GET Object tagging \(p. 368\)](#)

# SELECT Object Content

## Description

This operation filters the contents of an Amazon S3 object based on a simple structured query language (SQL) statement. In the request, along with the SQL expression, you must also specify a data serialization format (JSON, CSV, or Apache Parquet) of the object. Amazon S3 uses this format to parse object data into records, and returns only records that match the specified SQL expression. You must also specify the data serialization format for the response.

For more information about Amazon S3 Select, see [Selecting Content from Objects in the Amazon Simple Storage Service Developer Guide](#).

For more information about using SQL with Amazon S3 Select, see [SQL Reference for Amazon S3 Select and Glacier Select](#) in the [Amazon Simple Storage Service Developer Guide](#).

## Permissions

You must have `s3:GetObject` permission for this operation. Amazon S3 Select does not support anonymous access. For more information about permissions, see [Specifying Permissions in a Policy](#) in the [Amazon Simple Storage Service Developer Guide](#).

## Object Data Formats

You can use Amazon S3 Select to query objects that have the following format properties:

- **CSV, JSON, and Parquet** – Objects must be in CSV, JSON, or Parquet format.
- **UTF-8** – UTF-8 is the only encoding type Amazon S3 Select supports.
- **GZIP or BZIP2** – CSV and JSON files can be compressed using GZIP or BZIP2. GZIP and BZIP2 are the only compression formats that Amazon S3 Select supports for CSV and JSON files. Amazon S3 Select supports columnar compression for Parquet using GZIP or Snappy. Amazon S3 Select does not support whole-object compression for Parquet objects.
- **Server-side encryption** – Amazon S3 Select supports querying objects that are protected with server-side encryption.

For objects that are encrypted with customer-provided encryption keys (SSE-C), you must use HTTPS, and you must use the headers that are documented in the [Specific Request Headers for Server-Side Encryption with Customer-Provided Encryption Keys \(p. 353\)](#) section in the [Amazon S3 GET Object REST API](#). For more information about SSE-C, see [Server-Side Encryption \(Using Customer-Provided Encryption Keys\)](#) in the [Amazon Simple Storage Service Developer Guide](#).

For objects that are encrypted with Amazon S3 managed encryption keys (SSE-S3) and AWS KMS managed encryption keys (SSE-KMS), server-side encryption is handled transparently, so you don't need to specify anything. For more information about server-side encryption, including SSE-S3 and SSE-KMS, see [Protecting Data Using Server-Side Encryption](#) in the [Amazon Simple Storage Service Developer Guide](#).

## Requests

### Syntax

```
POST /ObjectName?select&select-type=2 HTTP/1.1
```

```
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (See Authenticating Requests \(AWS Signature Version 4\))
Request body goes here
```

#### Note

The syntax shows some of the request headers. For a complete list, see the "Request Headers" section of this topic.

Query parameters `select` and `select-type=2` are both required for all requests. `select-type=2` is present in order to enable extensions for future capabilities.

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Body

The following XML shows the request body for an object in CSV format with results in CSV format:

```
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from S3Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>GZIP</CompressionType>
        <CSV>
            <FileHeaderInfo>IGNORE</FileHeaderInfo>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,;</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
            <Comments>#</Comments>
            <AllowQuotedRecordDelimiter>FALSE</AllowQuotedRecordDelimiter>
        </CSV>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,;</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
    <RequestProgress>
        <Enabled>FALSE</Enabled>
    </RequestProgress>
</SelectRequest>
```

The following XML shows the request body for an object in JSON format with results in JSON format:

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

```

<Expression>Select * from S3Object</Expression>
<ExpressionType>SQL</ExpressionType>
<InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <JSON>
        <Type>DOCUMENT</Type>
    </JSON>
</InputSerialization>
<OutputSerialization>
    <JSON>
        <RecordDelimiter>\n</RecordDelimiter>
    </JSON>
</OutputSerialization>
<RequestProgress>
    <Enabled>FALSE</Enabled>
</RequestProgress>
</SelectRequest>

```

The following XML shows the request body for an object in Parquet format with results in CSV format:

```

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from S3Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>NONE</CompressionType>
        <Parquet>
        </Parquet>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
    <RequestProgress>
        <Enabled>FALSE</Enabled>
    </RequestProgress>
</SelectRequest>

```

### Note

In the XML:

- The `InputSerialization` element describes the format of the data in the object that is being queried. It must specify CSV, JSON, or Parquet.
- The `OutputSerialization` element describes the format of the data that you want Amazon S3 to return in response to the query. It must specify either CSV or JSON. Amazon S3 Select doesn't support outputting data in Parquet format.
- The format of the `InputSerialization` doesn't need to match the format of the `OutputSerialization`. So, for example, you can specify JSON in the `InputSerialization` and CSV in the `OutputSerialization`.

The following tables explain each of the XML elements in the request body.

Name	Description	Required
Expression	The SQL expression. For example:	Yes

Name	Description	Required
	<ul style="list-style-type: none"> <li>The following SQL expression retrieves the first column of the data from the object stored in CSV format.</li> </ul> <pre>SELECT s._1 FROM S3Object s</pre> <ul style="list-style-type: none"> <li>The following SQL expression returns everything from the object.</li> </ul> <pre>SELECT * FROM S3Object</pre> <p>Type: String Ancestor: SelectRequest</p>	
ExpressionType	Identifies the expression type.  Type: String  Valid values: SQL  Ancestor: SelectRequest	Yes
InputSerialization	Describes the format of the data in the object that is being queried.  Type: Container  Ancestor: SelectRequest	Yes
OutputSerialization	Describes the format of the data that you want Amazon S3 to return in response.  Type: Container  Ancestor: SelectRequest	Yes
RequestProgress	Describes optional, periodic QueryProgress messages that can be sent.  Type: Container  Ancestor: SelectRequest	No

### InputSerialization container element

Name	Description	Required
CompressionType	Identifies whether the Amazon S3 object that is being queried is compressed. GZIP and BZIP2 are the only supported compression types, and are supported only for CSV and JSON objects. If InputSerialization specifies the Parquet format, then CompressionType must be NONE, even if the Parquet object uses columnar compression.  Type: String  Valid values: NONE   GZIP   BZIP2	No

Name	Description	Required
	<p>Default: NONE</p> <p>Ancestor: <code>InputSerialization</code></p>	
<code>CSV   JSON   Parquet</code>	<p>Specifies the format and certain properties of the Amazon S3 object that is being queried.</p> <p>Type: Container</p> <p>Ancestor: <code>InputSerialization</code></p>	Exactly one of <code>CSV</code> , <code>JSON</code> , or <code>Parquet</code> is required.

### CSV container element (inside `InputSerialization`)

Name	Description	Required
<code>RecordDelimiter</code>	<p>The value used to separate individual records in the input. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, <code>\036</code> is parsed as the "record separator" (non-printing) character.</p> <p>You can specify up to two characters for a record delimiter. You can specify two characters, one character and one octal, or two octals. For example, <code>\r\n</code> is a valid record delimiter.</p> <p>Type: String</p> <p>Default: <code>\n</code></p> <p>Ancestor: <code>CSV</code></p>	No
<code>FieldDelimiter</code>	<p>The value used to separate individual fields in a record. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, <code>\036</code> is parsed as the "record separator" (non-printing) character.</p> <p>Type: String</p> <p>Default: <code>,</code></p> <p>Ancestor: <code>CSV</code></p>	No
<code>QuoteCharacter</code>	<p>The value to use for escaping when the field delimiter is part of the value.</p> <p>Consider this example in a CSV file:</p> <p><code>"a, b"</code></p> <p>The use of quotation marks makes this value a single field because you are wrapping the value in quotation marks. If you don't specify the quotation marks, the comma is a field delimiter (which makes it two separate field values, <code>a</code> and <code>b</code>).</p> <p>Type: String</p> <p>Default: <code>"</code></p>	No

Name	Description	Required
	Ancestor: CSV	
QuoteEscapeChar	The value to use for escaping the quotation mark character inside an already escaped value. For example, the value "" " a , b """ is parsed as " a , b ".  Type: String  Default: "  Ancestor: CSV	No
FileHeaderInfo	Describes the first line in the input data. It is one of the ENUM values.  • <b>NONE</b> : The first line is not a column header. • <b>USE</b> : The first line is a column header, and you can use the header value to identify a column in an expression (for example, <code>SELECT "name" FROM S3Object</code> ). • <b>IGNORE</b> : The first line is a column header, but you can't use the header values to identify the column in an expression. You can use column position (such as <code>_1, _2, ...</code> ) to identify the column (for example, <code>SELECT s._1 FROM S3Object s</code> ).  Type: Enum  Valid values: NONE   USE   IGNORE  Ancestor: CSV	No
Comments	If the first character of a line of text matches the comment character, the row is considered a comment and is discarded from the input. You can specify any character to indicate a comment line.  Type: String  Default: #  Ancestor: CSV	No
AllowQuotedRecords	Specifies that CSV input records might contain record delimiters within quote characters. Setting this option to TRUE could result in slower performance.  Type: Boolean  Default: FALSE  Ancestor: CSV	No

### JSON container element (inside InputSerialization)

Name	Description	Required
Type	<p>The type of JSON content. <code>LINES</code> means that each line in the input data contains a single JSON object. <code>DOCUMENT</code> means that a single JSON object can span multiple lines in the input. Using <code>DOCUMENT</code> might result in slower performance in some cases.</p> <p>Type: Enum</p> <p>Valid values: <code>DOCUMENT</code>   <code>LINES</code></p> <p>Ancestor: <code>JSON</code></p>	Yes

### OutputSerialization container element

Name	Description	Required
CSV   JSON	<p>Specifies the format and certain properties of the data that is returned in response.</p> <p>Type: Container</p> <p>Ancestor: <code>OutputSerialization</code></p>	Exactly one of <code>CSV</code> or <code>JSON</code> is required.

### CSV container element (inside OutputSerialization)

Name	Description	Required
QuoteFields	<p>Indicates whether to use quotation marks around output fields.</p> <ul style="list-style-type: none"> <li><code>ALWAYS</code>: Always use quotation marks for output fields.</li> <li><code>ASNEEDED</code>: Use quotation marks for output fields when needed.</li> </ul> <p>Type: String</p> <p>Valid values: <code>ALWAYS</code>   <code>ASNEEDED</code></p> <p>Default: <code>ASNEEDED</code></p> <p>Ancestor: <code>CSV</code></p>	No
RecordDelimiter	<p>The value used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, <code>\036</code> is parsed as the "record separator" (non-printing) character.</p> <p>You can specify up to two characters for a record delimiter. You can specify two characters, one character and one octal, or two octals. For example, <code>\r\n</code> is a valid record delimiter.</p> <p>Type: String</p> <p>Default: <code>\n</code></p>	No

Name	Description	Required
	Ancestor: CSV	
FieldDelimiter	<p>The value you want Amazon S3 to use to separate individual fields in a record. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, \\036 is parsed as the "record separator" (non-printing) character.</p> <p>Type: String</p> <p>Default: ,</p> <p>Ancestor: CSV</p>	No
QuoteCharacter	<p>The value to use for escaping when the field delimiter is part of the value. For example, if the value is a, b, then Amazon S3 wraps this field value in quotation marks as follows: " a , b ".</p> <p>Type: String</p> <p>Default: "</p> <p>Ancestor: CSV</p>	No
QuoteEscapeCharacter	<p>The value to use for escaping the quotation mark character inside an already escaped value. For example, if the value is " a , b ", then Amazon S3 wraps the value in quotation marks as follows: """ a , b """.</p> <p>Type: String</p> <p>Default: "</p> <p>Ancestor: CSV</p>	No

#### JSON container element (inside OutputSerialization)

Name	Description	Required
RecordDelimiter	<p>The value used to separate individual records in the output. Instead of the default value, you can specify an arbitrary delimiter, including an octal character. For example, \\036 is parsed as the "record separator" (non-printing) character.</p> <p>You can specify up to two characters for a record delimiter. You can specify two characters, one character and one octal, or two octals. For example, \r\n is a valid record delimiter.</p> <p>Type: String</p> <p>Default: \n</p> <p>Ancestor: JSON</p>	No

## RequestProgress container element

Name	Description	Required
Enabled	<p>Specifies whether periodic <code>QueryProgress</code> messages should be sent.</p> <p>Type: Boolean</p> <p>Default: <code>FALSE</code></p> <p>Ancestor: <code>RequestProgress</code></p>	No

# Responses

A successful operation returns 200 OK status code.

## Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

## Response Body

Because the response size is unknown, Amazon S3 streams the response as a series of messages and includes a `Transfer-Encoding` header with `chunked` as its value in the response. The following example shows the response format at the top level:

```
<Message 1>
<Message 2>
<Message 3>
.....
<Message n>
```

Each message consists of two sections: the prelude and the data. The prelude section consists of 1) the total byte-length of the message, and 2) the combined byte-length of all the headers. The data section consists of 1) the headers, and 2) a payload.

Each section ends with a 4-byte big-endian integer checksum (CRC). Amazon S3 Select uses CRC32 (often referred to as GZIP CRC32) to calculate both CRCs. For more information about CRC32, see [GZIP file format specification version 4.3](#).

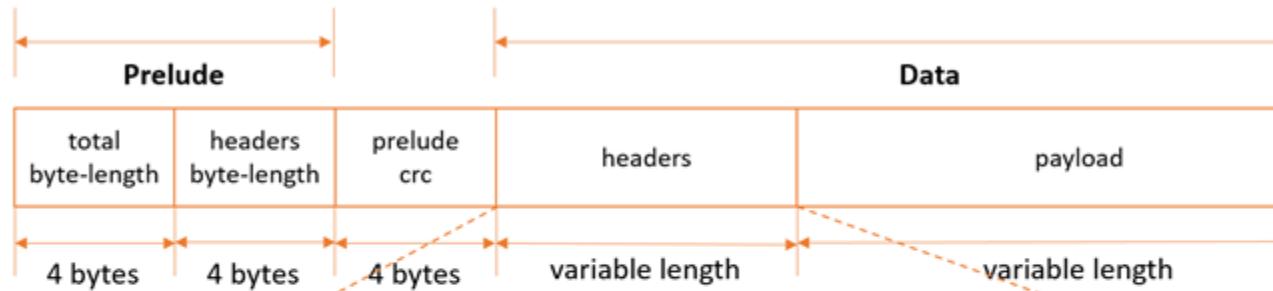
Total message overhead including the prelude and both checksums is 16 bytes.

### Note

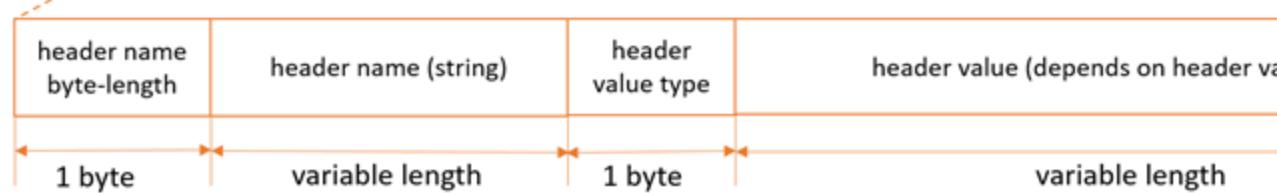
All integer values within messages are in network byte order, or big-endian order.

The following diagram shows the components that make up a message and a header. Note that there are multiple headers per message.

## Message:



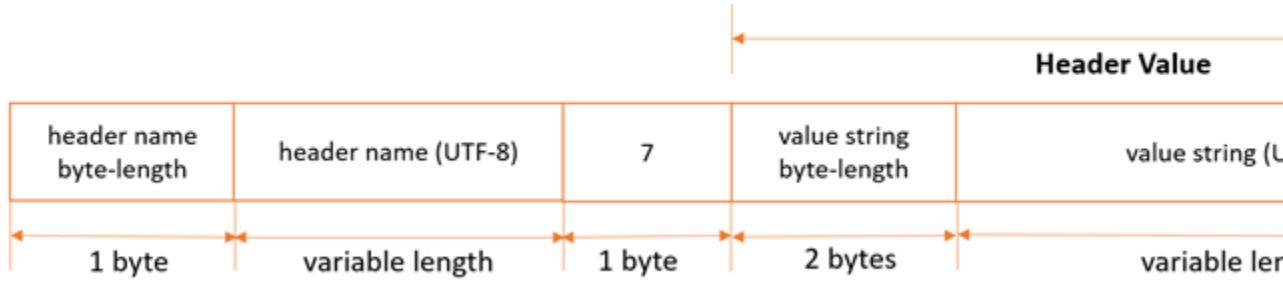
## Headers (multiple headers per message):



### Note

For Amazon S3 Select, the header value type is always 7 (type=String). For this type, the header value consists of two components, a 2-byte big-endian integer length, and a UTF-8 string that is of that byte-length. The following diagram shows the components that make up Amazon S3 Select headers.

## Amazon S3 Select Headers (type=String):



Payload byte-length calculations (these two calculations are equivalent):

- $\text{payload\_length} = \text{total\_length} - \text{header\_length} - \text{sizeOf(total\_length)} - \text{sizeOf(header\_length)} - \text{sizeOf(prelude\_crc)} - \text{sizeOf(message\_crc)}$
- $\text{payload\_length} = \text{total\_length} - \text{header\_length} - 16$

Each message contains the following components:

- **Prelude:** Always fixed size of 8 bytes (two fields of 4 bytes each):

- *First four bytes:* Total byte-length: Big-endian integer byte-length of the entire message (including the 4-byte total length field itself).
- *Second four bytes:* Headers byte-length: Big-endian integer byte-length of the headers portion of the message (excluding the headers length field itself).
- **Prelude CRC:** 4-byte big-endian integer checksum (CRC) for the prelude portion of the message (excluding the CRC itself). The prelude has a separate CRC from the message CRC (see below), to ensure that corrupted byte-length information can be detected immediately, without causing pathological buffering behavior.
- **Headers:** A set of metadata annotating the message, such as the message type, payload format, and so on. Messages can have multiple headers, so this portion of the message can have different byte-lengths depending on the message type. Headers are key-value pairs, where both the key and value are UTF-8 strings. Headers can appear in any order within the headers portion of the message, and any given header type can only appear once.

For Amazon S3 Select, following is a list of header names and the set of valid values depending on the message type.

- *MessageType Header:*
  - HeaderName => ":message-type"
  - Valid HeaderValues => "error", "event"
- *EventType Header:*
  - HeaderName => ":event-type"
  - Valid HeaderValues => "Records", "Cont", "Progress", "Stats", "End"
- *ErrorCode Header:*
  - HeaderName => ":error-code"
  - Valid HeaderValues => Error Code from the table in the [Special Errors \(p. 474\)](#) section.
- *ErrorMessage Header:*
  - HeaderName => ":error-message"
  - Valid HeaderValues => Error message returned by the service, to help diagnose request-level errors.
- **Payload:** Can be anything.
- **Message CRC:** 4-byte big-endian integer checksum (CRC) from the start of the message to the start of the checksum (that is, everything in the message excluding the message CRC itself).

Each header contains the following components. There can be multiple headers per message.

- **Header Name Byte-Length:** Byte-length of the header name.
- **Header Name:** Name of the header, indicating the header type. Valid values: ":message-type" ":event-type" ":error-code" ":error-message"
- **Header Value Type:** Enum indicating the header value type. For Amazon S3 Select, this is always 7.
- **Value String Byte-Length:** (For Amazon S3 Select) Byte-length of the header value string.
- **Header Value String:** (For Amazon S3 Select) Value of the header string. Valid values for this field vary based on the type of the header. See the sections below for valid values for each header type and message type.

For Amazon S3 Select, responses can be messages of the following types:

- **Records message:** Can contain a single record, partial records, or multiple records. Depending on the size of the result, a response can contain one or more of these messages.

- **Continuation message:** Amazon S3 periodically sends this message to keep the TCP connection open. These messages appear in responses at random. The client must detect the message type and process accordingly.
- **Progress message:** Amazon S3 periodically sends this message, if requested. It contains information about the progress of a query that has started but has not yet completed.
- **Stats message:** Amazon S3 sends this message at the end of the request. It contains statistics about the query.
- **End message:** Indicates that the request is complete, and no more messages will be sent. You should not assume that the request is complete until the client receives an End message.
- **RequestLevelError message:** Amazon S3 sends this message if the request failed for any reason. It contains the error code and error message for the failure. If Amazon S3 sends a RequestLevelError message, it doesn't send an End message.

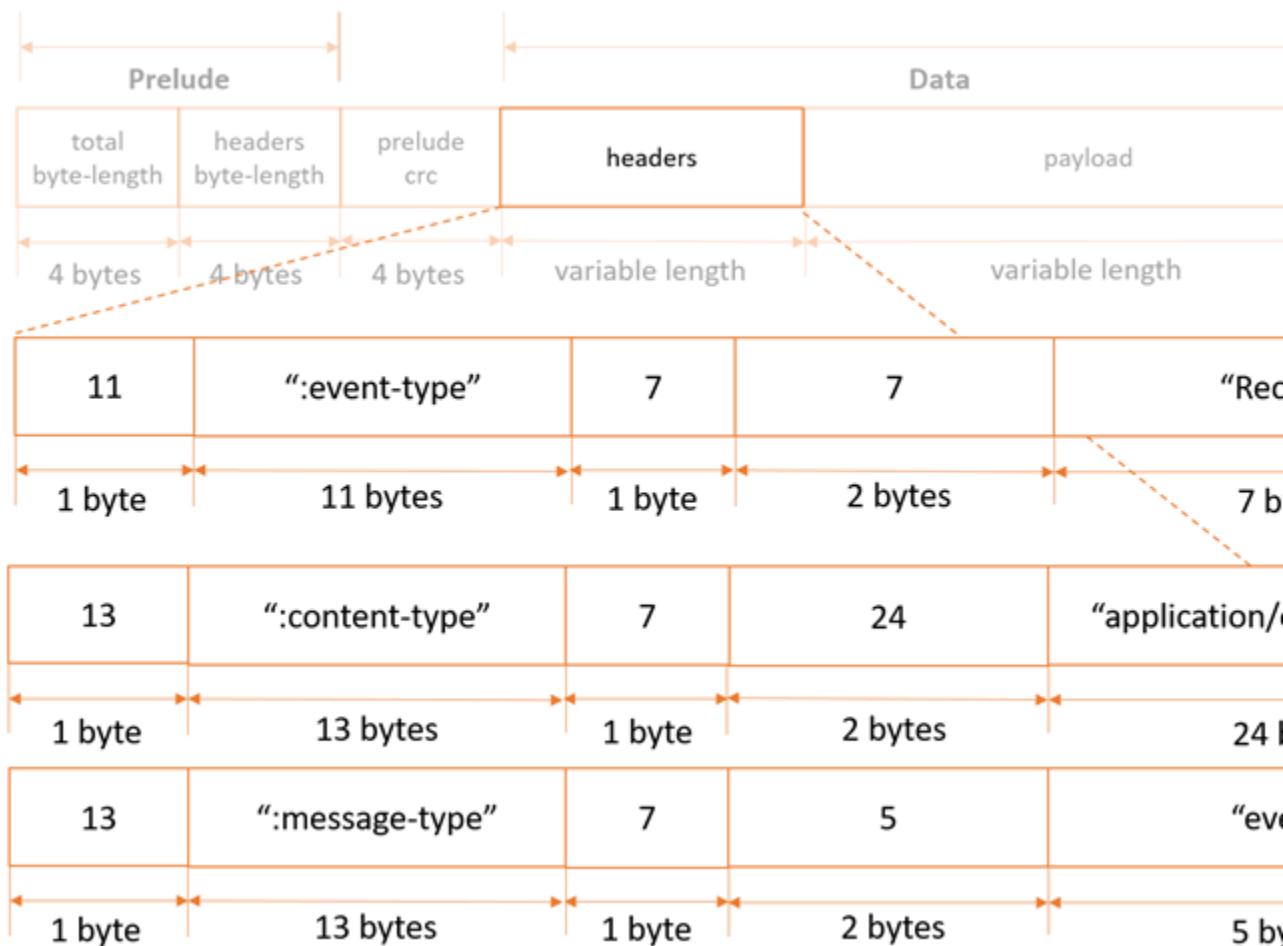
The following sections explain the structure of each message type in more detail.

For sample code and unit tests that use this protocol, see [AWS C Event Stream](#) on the GitHub website.

## Records Message

### Header specification

Records messages contain three headers, as follows:



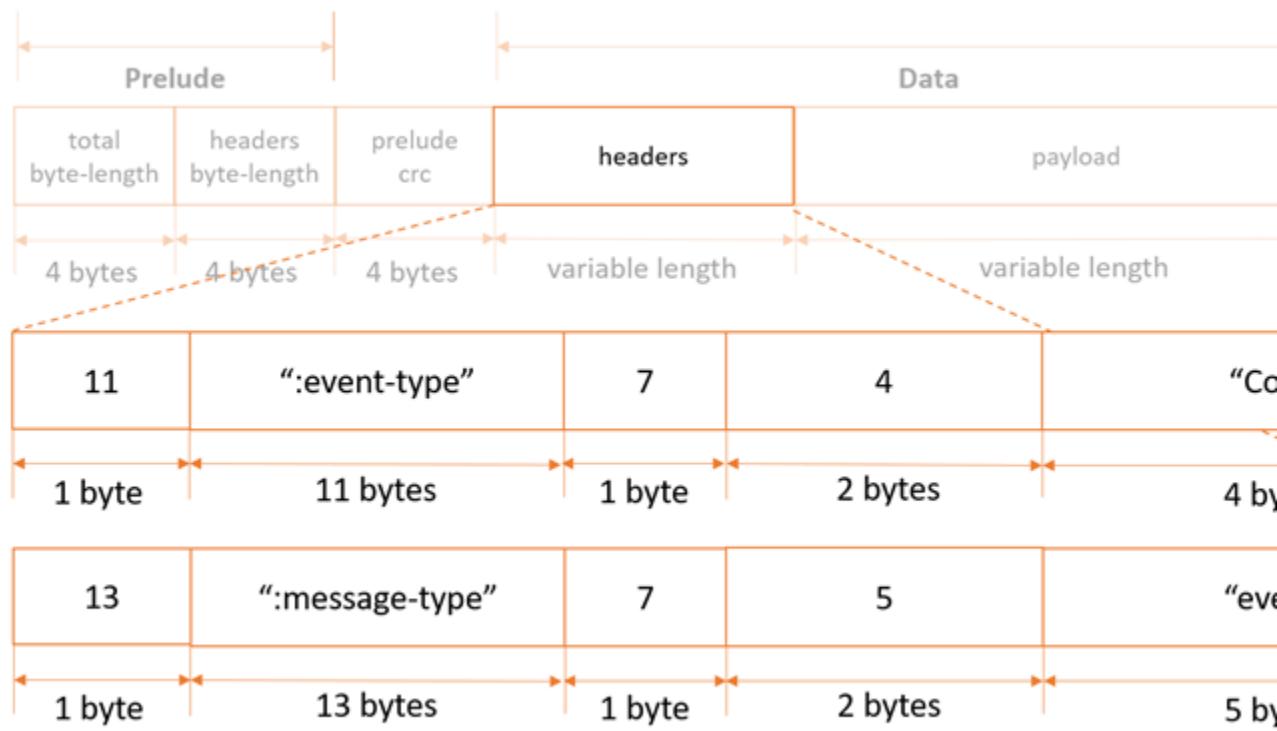
### [Payload specification](#)

Records message payloads can contain a single record, partial records, or multiple records.

### [Continuation Message](#)

#### [Header specification](#)

Continuation messages contain two headers, as follows:



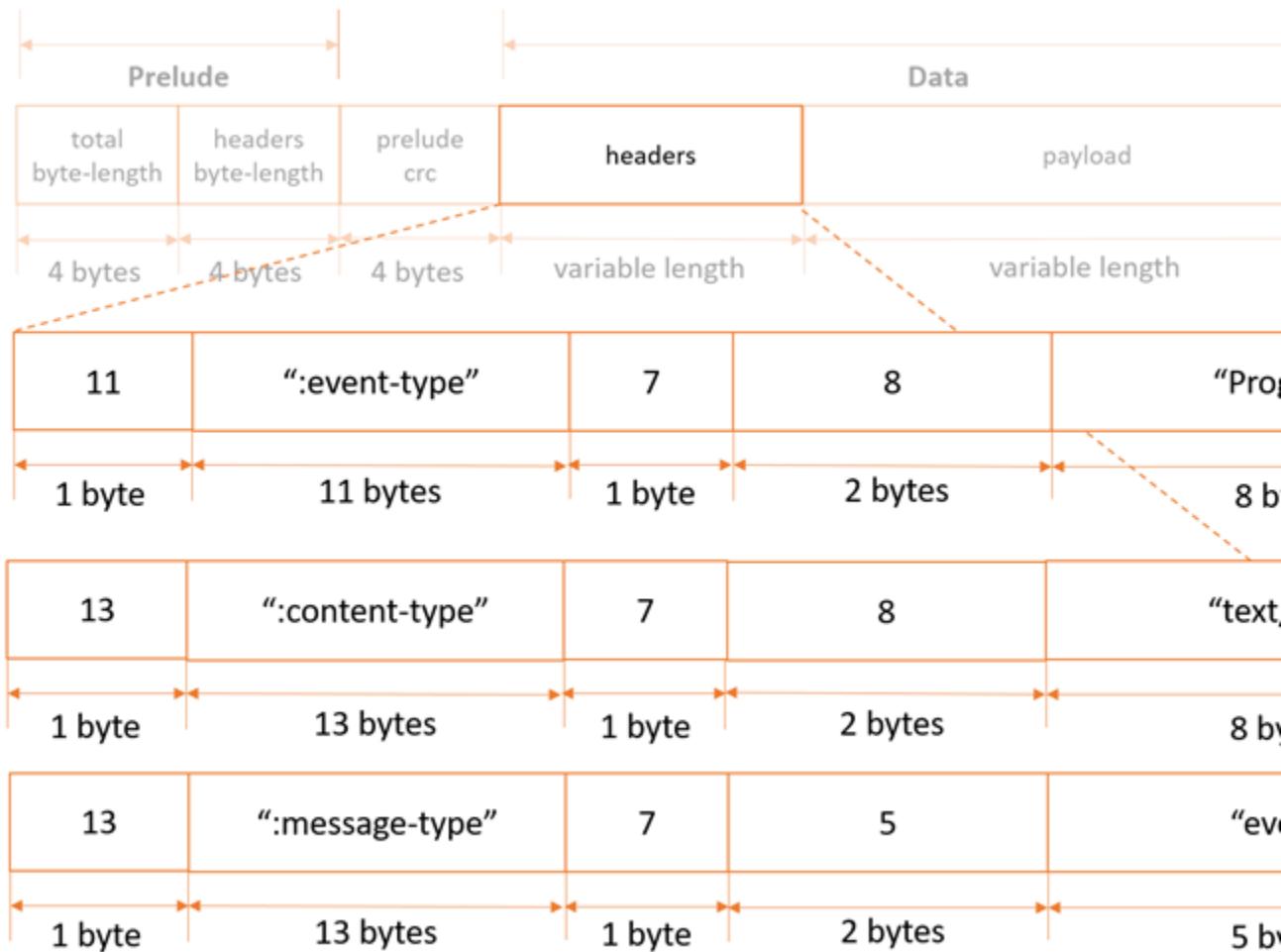
### [Payload specification](#)

Continuation messages have no payload.

### [Progress Message](#)

#### [Header specification](#)

Progress messages contain three headers, as follows:



### Payload specification

Progress message payload is an XML document containing information about the progress of a request.

- *BytesScanned* => Number of bytes that have been processed before being uncompressed (if the file is compressed).
- *BytesProcessed* => Number of bytes that have been processed after being uncompressed (if the file is compressed).
- *BytesReturned* => Current number of bytes of records payload data returned by Amazon S3.

For uncompressed files, *BytesScanned* and *BytesProcessed* are equal.

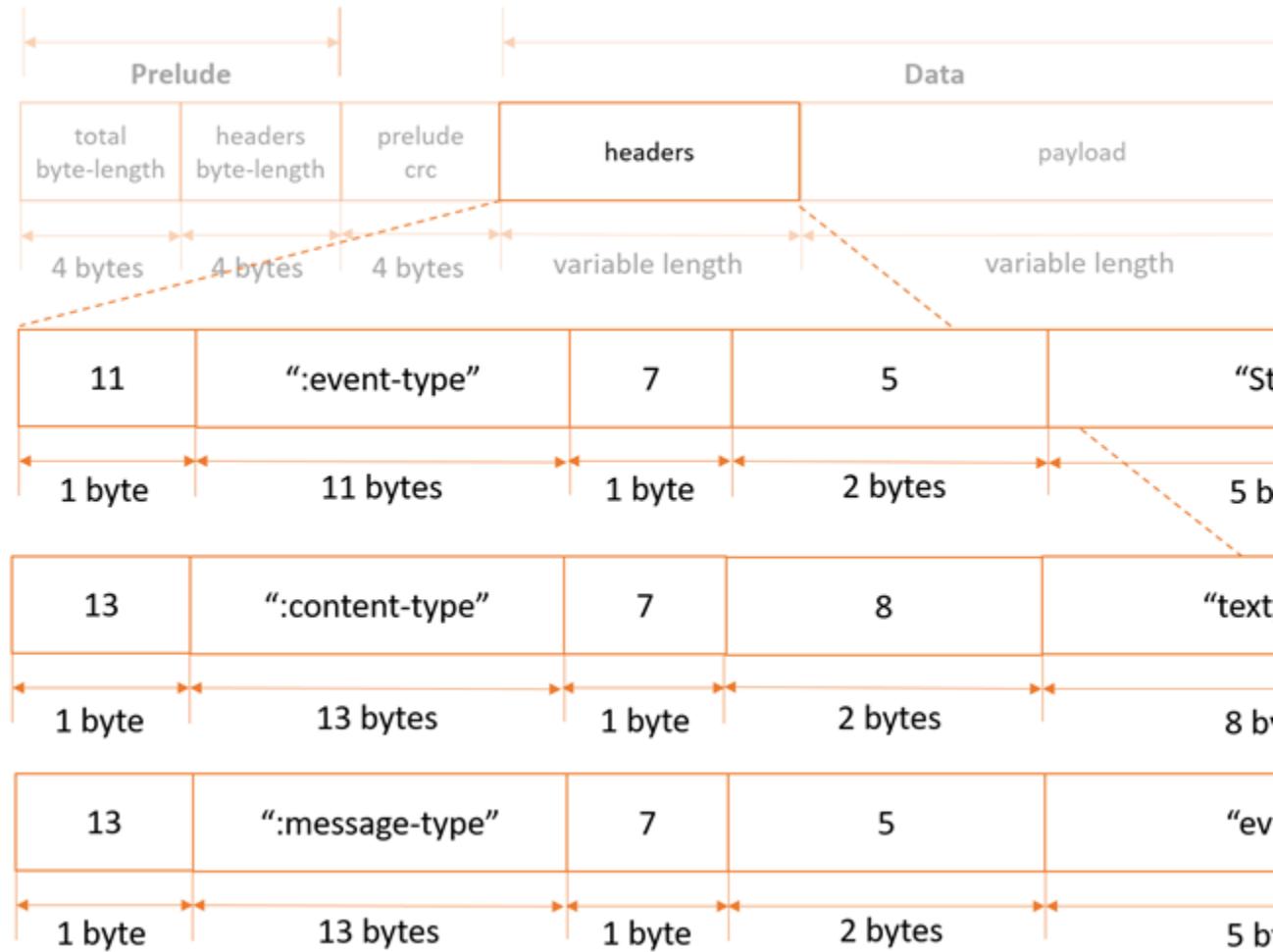
Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Progress>
    <BytesScanned>512</BytesScanned>
    <BytesProcessed>1024</BytesProcessed>
    <BytesReturned>1024</BytesReturned>
</Progress>
```

## Stats Message

### Header specification

Stats messages contain three headers, as follows:



### Payload specification

Stats message payload is an XML document containing information about a request's stats when processing is complete.

- BytesScanned* => Number of bytes that have been processed before being uncompressed (if the file is compressed).
- BytesProcessed* => Number of bytes that have been processed after being uncompressed (if the file is compressed).
- BytesReturned* => Total number of bytes of records payload data returned by Amazon S3.

For uncompressed files, *BytesScanned* and *BytesProcessed* are equal.

Example:

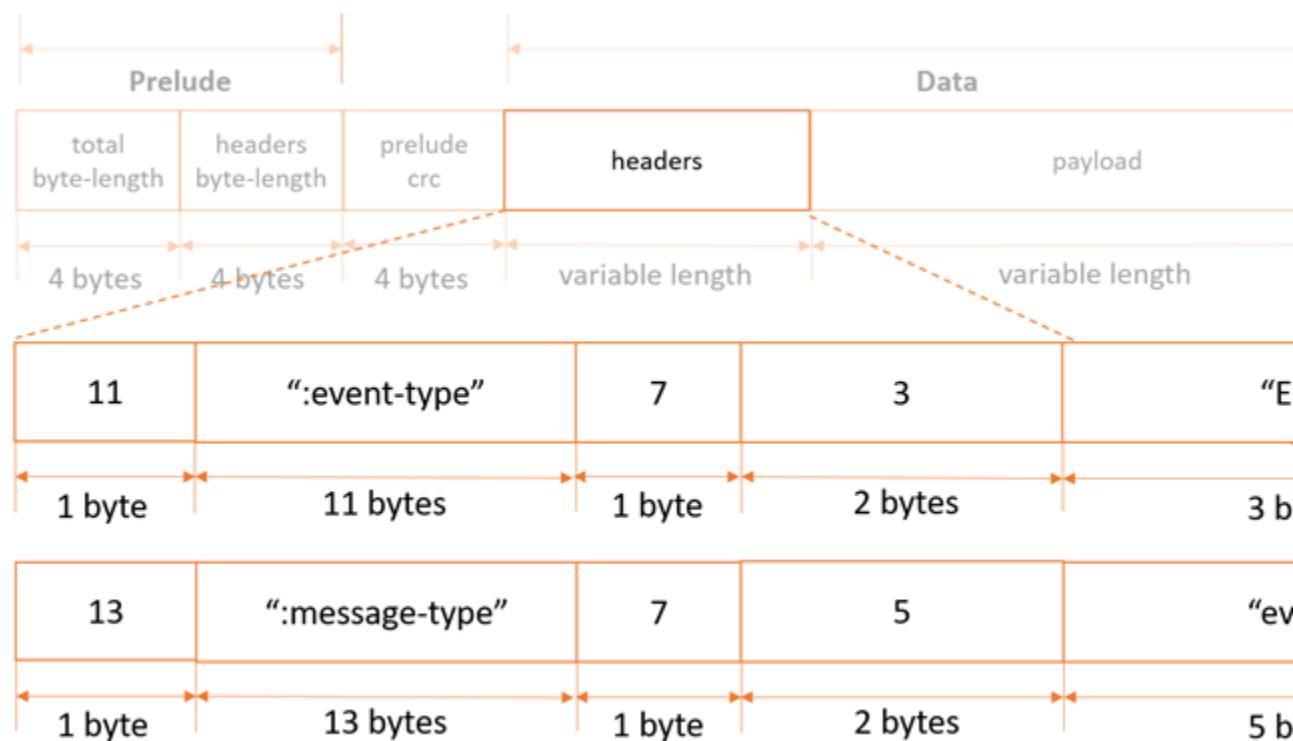
```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Stats>
<BytesScanned>512</BytesScanned>
<BytesProcessed>1024</BytesProcessed>
<BytesReturned>1024</BytesReturned>
</Stats>
```

## End Message

### Header specification

End messages contain two headers, as follows:



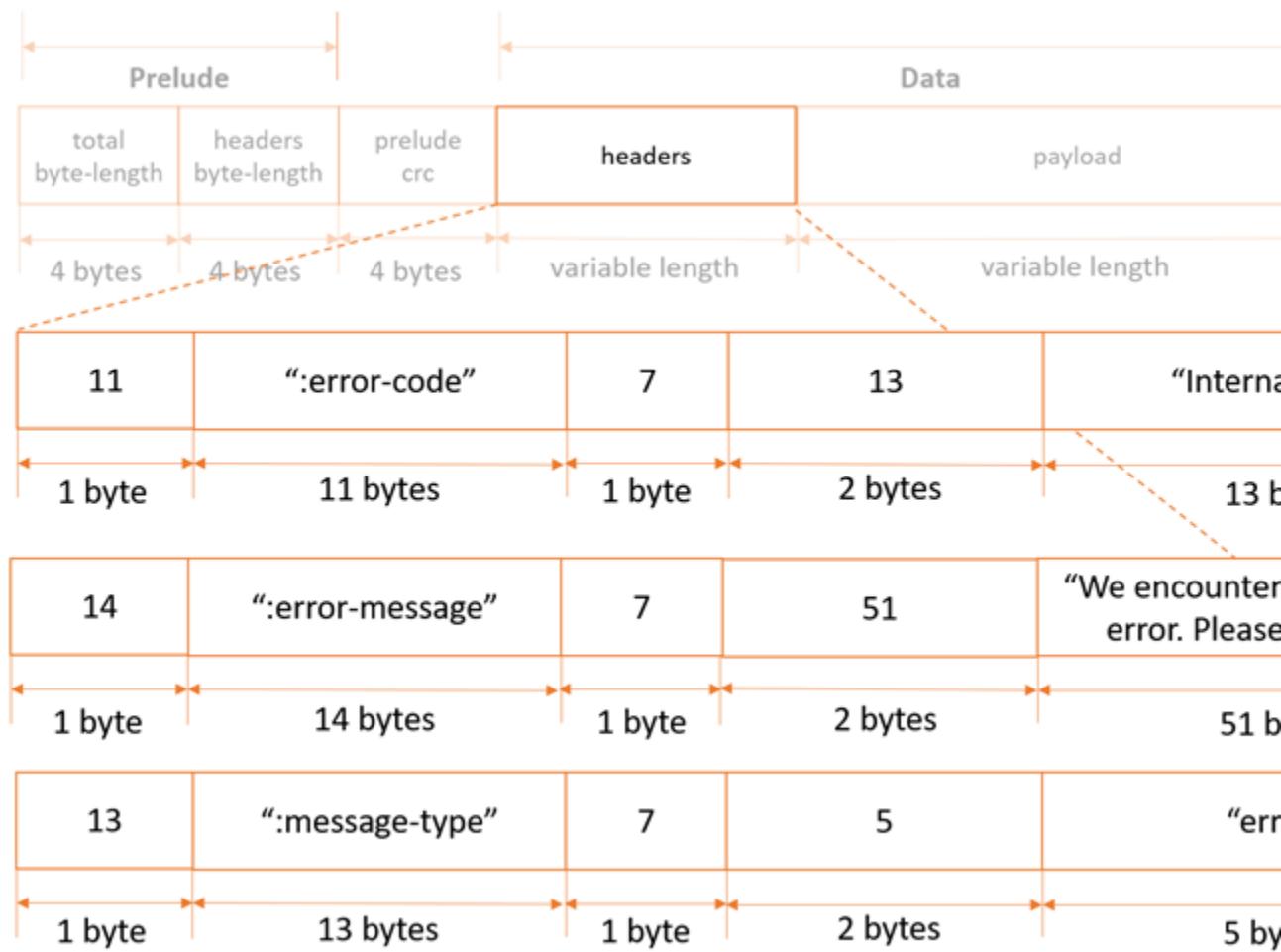
### Payload specification

End messages have no payload.

## Request Level Error Message

### Header specification

Request-level error messages contain three headers, as follows:



For a list of possible error codes and error messages, see the table in the [Special Errors \(p. 474\)](#) section.

#### [Payload specification](#)

Request-level error messages have no payload.

## Special Errors

The following table contains special errors that `SELECT Object Content` might return.

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
Busy	The service is unavailable. Please retry.	503	Client
UnauthorizedAccess	You are not authorized to perform this operation	401	Client
EmptyRequestBody	Request body cannot be empty.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
ExpressionTooLong	The SQL expression is too long: The maximum byte-length for the SQL expression is 256 KB.	400	Client
IllegalSqlFunctionArgument	Illegal argument was used in the SQL function.	400	Client
InternalError	Encountered an internal error.	500	Client
InvalidColumnIndex	Column index in the SQL expression is invalid.	400	Client
InvalidKeyPath	Key path in the SQL expression is invalid.	400	Client
ColumnTooLong	The length of a column in the result is greater than maxCharsPerColumn of 1 MB.	400	Client
OverMaxColumn	The number of columns in the result is greater than the maximum allowable number of columns.	400	Client
OverMaxRecordSize	The length of a record in the input or result is greater than maxCharsPerRecord of 1 MB.	400	Client
MissingHeaders	Some headers in the query are missing from the file. Check the file and try again.	400	Client
InvalidCompressionFormat	The file is not in a supported compression format. Only GZIP and BZIP2 are supported.	400	Client
TruncatedInput	Object decompression failed. Check that the object is properly compressed using the format specified in the request.	400	Client
InvalidExpressionType	The ExpressionType is invalid. Only SQL expressions are supported.	400	Client
InvalidFileInfo	The FileInfo is invalid. Only NONE, USE, and IGNORE are supported.	400	Client
InvalidJsonType	The JsonType is invalid. Only DOCUMENT and LINES are supported.	400	Client
InvalidQuoteFields	The QuoteFields is invalid. Only ALWAYS and ASNEEDED are supported.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
InvalidRequestParameter	The value of a parameter in SelectRequest element is invalid. Check the service API documentation and try again.	400	Client
CSVParsingError	Encountered an error parsing the CSV file. Check the file and try again.	400	Client
JSONParsingError	Encountered an error parsing the JSON file. Check the file and try again.	400	Client
ExternalEvalException	The query cannot be evaluated. Check the file and try again.	400	Client
InvalidDataType	The SQL expression contains an invalid data type.	400	Client
UnrecognizedFormatException	Encountered an invalid record type.	400	Client
InvalidTextEncoding	Invalid encoding type. Only UTF-8 encoding is supported.	400	Client
InvalidDataSource	Invalid data source type. Only CSV, JSON, and Parquet are supported.	400	Client
InvalidTableAlias	The SQL expression contains an invalid table alias.	400	Client
MalformedXML	The XML provided was not well-formed or did not validate against our published schema. Check the service documentation and try again.	400	Client
MultipleDataSourcesUnsupported	Multiple data sources are not supported.	400	Client
MissingRequiredParameter	The SelectRequest entity is missing a required parameter. Check the service documentation and try again.	400	Client
ObjectSerializationConflict	InputSerialization specifies more than one format (CSV, JSON, or Parquet), or OutputSerialization specifies more than one format (CSV or JSON). InputSerialization and OutputSerialization can only specify one format each.	400	Client
UnsupportedFunction	Encountered an unsupported SQL function.	400	Client
UnsupportedSqlOperation	Encountered an unsupported SQL operation.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
UnsupportedSqlStructure	Encountered an unsupported SQL structure. Check the SQL Reference.	400	Client
UnsupportedStorageClass	Encountered an invalid storage class. Only STANDARD, STANDARD_IA, and ONEZONE_IA storage classes are supported.	400	Client
UnsupportedSyntax	Encountered invalid syntax.	400	Client
UnsupportedRangeHeader	Range header is not supported for this operation.	400	Client
LexerInvalidChar	The SQL expression contains an invalid character.	400	Client
LexerInvalidOperator	The SQL expression contains an invalid literal.	400	Client
LexerInvalidLiteral	The SQL expression contains an invalid operator.	400	Client
LexerInvalidIONLiteral	The SQL expression contains an invalid operator.	400	Client
ParseExpectedDatePart	Did not find the expected date part in the SQL expression.	400	Client
ParseExpectedKeyword	Did not find the expected keyword in the SQL expression.	400	Client
ParseExpectedTokenType	Did not find the expected token in the SQL expression.	400	Client
ParseExpected2TokenTypes	Did not find the expected token in the SQL expression.	400	Client
ParseExpectedNumber	Did not find the expected number in the SQL expression.	400	Client
ParseExpectedRightParenBuild	Did not find the expected right parenthesis character in the SQL expression.	400	Client
ParseExpectedTypeName	Did not find the expected type name in the SQL expression.	400	Client
ParseExpectedWhenClause	Did not find the expected WHEN clause in the SQL expression. CASE is not supported.	400	Client
ParseUnsupportedToken	The SQL expression contains an unsupported token.	400	Client
ParseUnsupportedLiteralsGroupBy	The SQL expression contains an unsupported use of GROUP BY.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
ParseExpectedMember	The SQL expression contains an unsupported use of MEMBER.	400	Client
ParseUnsupportedSelect	The SQL expression contains an unsupported use of SELECT.	400	Client
ParseUnsupportedCase	The SQL expression contains an unsupported use of CASE.	400	Client
ParseUnsupportedCaseClause	The SQL expression contains an unsupported use of CASE.	400	Client
ParseUnsupportedAlias	The SQL expression contains an unsupported use of ALIAS.	400	Client
ParseUnsupportedSyntax	The SQL expression contains unsupported syntax.	400	Client
ParseUnknownOperator	The SQL expression contains an invalid operator.	400	Client
ParseInvalidPathComponent	The SQL expression contains an invalid path component.	400	Client
ParseMissingIdentAfterAt	Did not find the expected identifier after the @ symbol in the SQL expression.	400	Client
ParseUnexpectedOperator	The SQL expression contains an unexpected operator.	400	Client
ParseUnexpectedTerm	The SQL expression contains an unexpected term.	400	Client
ParseUnexpectedToken	The SQL expression contains an unexpected token.	400	Client
ParseUnExpectedKeyword	The SQL expression contains an unexpected keyword.	400	Client
ParseExpectedExpression	Did not find the expected SQL expression.	400	Client
ParseExpectedLeftParenAfterCast	Did not find the expected left parenthesis after CAST in the SQL expression.	400	Client
ParseExpectedLeftParenValue	Did not find the expected left parenthesis in the SQL expression.	400	Client
ParseExpectedLeftParenBuild	Did not find the expected left parenthesis in the SQL expression.	400	Client
ParseExpectedArgumentDelimiter	Did not find the expected argument delimiter in the SQL expression.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
ParseCastArity	The SQL expression CAST has incorrect arity.	400	Client
ParseInvalidTypeParam	The SQL expression contains an invalid parameter value.	400	Client
ParseEmptySelect	The SQL expression contains an empty SELECT.	400	Client
ParseSelectMissingFrom	The SQL expression contains a missing FROM after SELECT list.	400	Client
ParseExpectedIdentForGroupBy	GROUP BY is not supported in the SQL expression.	400	Client
ParseExpectedIdentForAlias	Did not find the expected identifier for the alias in the SQL expression.	400	Client
ParseUnsupportedCallWithStar	Only COUNT with (*) as a parameter is supported in the SQL expression.	400	Client
ParseNonUnaryAggregateFunction	Only one argument is supported for aggregate functions in the SQL expression.	400	Client
ParseMalformedJoin	JOIN is not supported in the SQL expression.	400	Client
ParseExpectedIdentForAt	Did not find the expected identifier for AT name in the SQL expression.	400	Client
ParseAsteriskIsNotAloneInSelectExpression	Other expressions are not allowed in the SELECT list when '*' is used without dot notation in the SQL expression.	400	Client
ParseCannotMixSqbAndWildcardsInSelectList	Cannot mix [ ] and * in the same expression in a SELECT list in SQL expression.	400	Client
ParseInvalidContextForWildcards	Cannot use wildcards in the same SELECT list in the SQL expression.	400	Client
EvaluatorBindingDoesNotExist	A column name or a path provided does not exist in the SQL expression.	400	Client
ValueParseFailure	Time stamp parse failure in the SQL expression.	400	Client
IncorrectSqlFunctionArguments	Incorrect type of arguments in function call in the SQL expression.	400	Client
AmbiguousFieldName	Field name matches to multiple fields in the file. Check the SQL expression and the file, and try again.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
EvaluatorInvalidArguments	Incorrect number of arguments in the function call in the SQL expression.	400	Client
EvaluatorInvalidTimestamp	Invalid time stamp format string in the SQL expression.	400	Client
ValueParseFailure	Time stamp parse failure in the SQL expression.	400	Client
IntegerOverflow	Integer overflow or underflow in the SQL expression.	400	Client
LikeInvalidInputs	Invalid argument given to the LIKE clause in the SQL expression.	400	Client
CastFailed	Attempt to convert from one data type to another using CAST failed in the SQL expression.	400	Client
InvalidCast	Attempt to convert from one data type to another using CAST failed in the SQL expression.	400	Client
EvaluatorInvalidTimestampFormat	Time stamp format pattern requires additional fields in the SQL expression.	400	Client
EvaluatorInvalidTimestampFormatPattern	Time stamp format pattern contains a valid format symbol that cannot be applied to time stamp parsing in the SQL expression.	400	Client
EvaluatorTimestampFormatPatternMultiple	Time stamp format pattern contains multiple format specifiers representing the time stamp field in the SQL expression.	400	Client
EvaluatorTimestampFormatPatternTwelveHour	Time stamp format pattern contains a 12-hour hour of day format symbol but doesn't also contain an AM/PM field, or it contains a 24-hour hour of day format specifier and contains an AM/PM field in the SQL expression.	400	Client
EvaluatorUnterminatedTimestampFormatPattern	Time stamp format pattern contains unterminated token in the SQL expression.	400	Client
EvaluatorInvalidTimestampFormatPattern	Time stamp format pattern contains an invalid token in the SQL expression.	400	Client

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
EvaluatorInvalidTimestamp	Timestamp format pattern contains an invalid symbol in the SQL expression.	400	Client
ParquetParsingError	Error parsing Parquet file. Please check the file and try again.	400	Client

## Examples

### Example 1: CSV Object

The following select request retrieves all records from an object with data stored in CSV format. The `OutputSerialization` element directs Amazon S3 to return results in CSV.

```
POST /exampleobject.csv?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from S3Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>GZIP</CompressionType>
        <CSV>
            <FileHeaderInfo>IGNORE</FileHeaderInfo>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
            <Comments>#</Comments>
        </CSV>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
</SelectRequest>
```

You can try different queries in the `Expression` element:

- Assuming that you are not using column headers, you can identify columns using positional headers:

```
SELECT s._1, s._2 FROM S3Object s WHERE s._3 > 100
```

- If you have column headers and you set the `FileHeaderInfo` to `Use`, you can identify columns by name in the expression:

```
SELECT s.Id, s.FirstName, s.SSN FROM S3Object s
```

- You can specify functions in the SQL expression:

```
SELECT count(*) FROM S3Object s WHERE s._1 < 1
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: GFihv3y6+kE7KG11GEkQhU7/2/cHR3Yb2fCb2S04nxI423Dqwg2XiQ0B/UzlzYQvPiBlZNRcovw=
x-amz-request-id: 9F341CD3C4BA79E0
Date: Tue, 17 Oct 2017 23:54:05 GMT

A series of messages
```

## Example 2: JSON Object

The following select request retrieves all records from an object with data stored in JSON format. The `OutputSerialization` directs Amazon S3 to return results in CSV.

```
POST /exampleobject.json?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from S3Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>GZIP</CompressionType>
        <JSON>
            <Type>DOCUMENT</Type>
        </JSON>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
</SelectRequest>
```

You can try different queries in the `Expression` element:

- You can filter by string comparison using record keys:

```
SELECT s.country, s.city from S3Object s where s.city = 'Seattle'
```

- You can specify functions in the SQL expression:

```
SELECT count(*) FROM S3Object s
```

## Example 3: Parquet Object

The following select request retrieves all records from an object with data stored in Parquet format. The `OutputSerialization` directs Amazon S3 to return results in CSV.

```
POST /exampleobject.parquet?select&select-type=2 HTTP/1.1
Host: examplebucket.s3.amazonaws.com
Date: Tue, 17 Oct 2017 01:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
    <Expression>Select * from S3Object</Expression>
    <ExpressionType>SQL</ExpressionType>
    <InputSerialization>
        <CompressionType>NONE</CompressionType>
        <Parquet>
        </Parquet>
    </InputSerialization>
    <OutputSerialization>
        <CSV>
            <QuoteFields>ASNEEDED</QuoteFields>
            <RecordDelimiter>\n</RecordDelimiter>
            <FieldDelimiter>,</FieldDelimiter>
            <QuoteCharacter>"</QuoteCharacter>
            <QuoteEscapeCharacter>"</QuoteEscapeCharacter>
        </CSV>
    </OutputSerialization>
</SelectRequest>
```

## Notes

The `SELECT Object Content` operation does not support the following `GET Object` functionality. For more information, see [GET Object \(p. 349\)](#).

- **Range:** You cannot specify the range of bytes of an object to return.
- **GLACIER and REDUCED\_REDUNDANCY storage classes:** You cannot specify either the `GLACIER` or `REDUCED_REDUNDANCY` storage classes. For more information, about storage classes see [Storage Classes](#) in the *Amazon Simple Storage Service Developer Guide*.

## Related Resources

- [GET Object \(p. 349\)](#)
- [GET Bucket lifecycle \(p. 145\)](#)
- [PUT Bucket lifecycle \(p. 265\)](#)

# Abort Multipart Upload

## Description

This operation aborts a multipart upload. After a multipart upload is aborted, no additional parts can be uploaded using that upload ID. The storage consumed by any previously uploaded parts will be freed. However, if any part uploads are currently in progress, those part uploads might or might not succeed. As a result, it might be necessary to abort a given multipart upload multiple times in order to completely free all storage consumed by all parts. To verify that all parts have been removed, so you don't get charged for the part storage, you should call the [List Parts \(p. 502\)](#) operation and ensure the parts list is empty.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
DELETE /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

This operation uses only Request Headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This operation does not use request elements.

## Responses

### Response Headers

This operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This operation does not use response elements.

## Special Errors

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
NoSuchUpload	The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.	404 Not Found	Client

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following request aborts a multipart upload identified by its upload ID.

```
DELETE /example-object?uploadId=VXBsb2FkIE1EIGZvcibLbHZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hz
HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 204 OK
x-amz-id-2: Weag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 996c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 0
Connection: keep-alive
Server: AmazonS3
```

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)
- [Upload Part \(p. 508\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [List Parts \(p. 502\)](#)
- [List Multipart Uploads \(p. 218\)](#)

# Complete Multipart Upload

## Description

This operation completes a multipart upload by assembling previously uploaded parts.

You first initiate the multipart upload and then upload all parts using the Upload Parts operation (see [Upload Part \(p. 508\)](#)). After successfully uploading all relevant parts of an upload, you call this operation to complete the upload. Upon receiving this request, Amazon S3 concatenates all the parts in ascending order by part number to create a new object. In the Complete Multipart Upload request, you must provide the parts list. You must ensure the parts list is complete, this operation concatenates the parts you provide in the list. For each part in the list, you must provide the part number and the `ETag` header value, returned after that part was uploaded.

Processing of a Complete Multipart Upload request could take several minutes to complete. After Amazon S3 begins processing the request, it sends an HTTP response header that specifies a `200 OK` response. While processing is in progress, Amazon S3 periodically sends whitespace characters to keep the connection from timing out. Because a request could fail after the initial `200 OK` response has been sent, it is important that you check the response body to determine whether the request succeeded.

Note that if Complete Multipart Upload fails, applications should be prepared to retry the failed requests. For more information, go to [Amazon S3 Error Best Practices](#) section of the *Amazon Simple Storage Service Developer Guide*.

For more information on multipart uploads, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon Simple Storage Service Developer Guide*.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
POST /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Content-Length: Size
Authorization: authorization string

<CompleteMultipartUpload>
  <Part>
    <PartNumber>PartNumber</PartNumber>
    <ETag>ETag</ETag>
  </Part>
  ...
</CompleteMultipartUpload>
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

This operation uses only Request Headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#)

## Request Elements

Name	Description	Required
CompleteMultipartUpload	Container for the request.  Ancestor: None  Type: Container  Children: One or more <b>Part</b> elements	Yes
Part	Container for elements related to a particular previously uploaded part.  Ancestor: <b>CompleteMultipartUpload</b>  Type: Container  Children: <b>PartNumber</b> , <b>ETag</b>	Yes
PartNumber	Part number that identifies the part.  Ancestor: <b>Part</b>  Type: Integer	Yes
ETag	Entity tag returned when the part was uploaded.  Ancestor: <b>Part</b>  Type: String	Yes

## Responses

### Response Headers

The operation uses the following response header, in addition to the response headers common to most requests. For more information, see [Common Response Headers \(p. 4\)](#).

Header	Description
x-amz-expiration	Amazon S3 returns this header if an <a href="#">Expiration</a> action is configured for the object as part of the bucket's lifecycle configuration. The header value includes an "expiry-date" component and a URL-encoded "rule-id" component. Note that for versioning-enabled buckets, this header applies only to current versions; Amazon S3 does not provide a header to infer when a noncurrent version will be eligible for permanent deletion. For more information, see <a href="#">PUT Bucket lifecycle (p. 265)</a> .  Type: String
x-amz-server-side-encryption	If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.

Header	Description
	Type: String
x-amz-server-side-encryption-aws-kms-key-id	If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code> , this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.  Type: String
x-amz-server-side-encryption-customer-algorithm	If encryption by using server-side encryption with customer-provided encryption keys was requested, the response will include this header confirming the encryption algorithm used.  Type: String  Valid Value: AES256
x-amz-version-id	Version ID of the newly created object, in case the bucket has versioning turned on.  Type: String

## Response Elements

Name	Description
CompleteMultipartUploadResult	Container for the response  Type: Container  Children: Location, Bucket, Key, ETag  Ancestors: None
Location	The URI that identifies the newly created object.  Type: URI  Ancestors: CompleteMultipartUploadResult
Bucket	The name of the bucket that contains the newly created object.  Type: String  Ancestors: CompleteMultipartUploadResult
Key	The object key of the newly created object.  Type: String  Ancestors: CompleteMultipartUploadResult
ETag	Entity tag that identifies the newly created object's data. Objects with different object data will have different entity tags. The entity tag is an opaque string. The entity tag may or may not be an MD5 digest of the object data. If the entity tag is not an MD5 digest of the object data, it will

Name	Description
	contain one or more nonhexadecimal characters and/or will consist of less than 32 or more than 32 hexadecimal digits.  Type: String  Ancestors: CompleteMultipartUploadResult

## Special Errors

Error Code	Description	HTTP Status Code
EntityTooSmall	Your proposed upload is smaller than the minimum allowed object size. Each part must be at least 5 MB in size, except the last part.	400 Bad Request
InvalidPart	One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag.	400 Bad Request
InvalidPartOrder	The list of parts was not in ascending order. The parts list must be specified in order by part number.	400 Bad Request
NoSuchUpload	The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.	404 Not Found

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following Complete Multipart Upload request specifies three parts in the CompleteMultipartUpload element.

```
POST /example-object?uploadId=AAAsb2FkIE1EIGZvciB1bHZpbmcncyWeeS1tb3ZpZS5tMnRzIRRwbG9hZA
HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 391
Authorization: authorization string

<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"a54357aff0632cce46d942af68356b38"</ETag>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <ETag>"0c78aef83f66abc1fa1e8477f296d394"</ETag>
  </Part>
  <Part>
    <PartNumber>3</PartNumber>
```

```
<ETag>"acbd18db4cc2f85cedef654fccc4a4d8"</ETag>
</Part>
</CompleteMultipartUpload>
```

## Sample Response

The following response indicates that an object was successfully assembled.

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOfg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Location>http://Example-Bucket.s3.amazonaws.com/Example-Object</Location>
  <Bucket>Example-Bucket</Bucket>
  <Key>Example-Object</Key>
  <ETag>"3858f62230ac3c915f300c664312c11f-9"</ETag>
</CompleteMultipartUploadResult>
```

## Sample Response with Error Specified in Header

The following response indicates that an error occurred before the HTTP response header was sent.

```
HTTP/1.1 403 Forbidden
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOfg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 237
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>656c76696e6727732072657175657374</RequestId>
  <HostId>Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOfg==</HostId>
</Error>
```

## Sample Response with Error Specified in Body

The following response indicates that an error occurred after the HTTP response header was sent. Note that while the HTTP status code is 200 OK, the request actually failed as described in the **Error** element.

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOfg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Connection: close
Server: AmazonS3

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

<Error>
  <Code>InternalError</Code>
```

```
<Message>We encountered an internal error. Please try again.</Message>
<RequestId>656c76696e6727732072657175657374</RequestId>
<HostId>Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFG==</HostId>
</Error>
```

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)
- [Upload Part \(p. 508\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Parts \(p. 502\)](#)
- [List Multipart Uploads \(p. 218\)](#)

# Initiate Multipart Upload

## Description

This operation initiates a multipart upload and returns an upload ID. This upload ID is used to associate all of the parts in the specific multipart upload. You specify this upload ID in each of your subsequent upload part requests (see [Upload Part \(p. 508\)](#)). You also include this upload ID in the final request to either complete or abort the multipart upload request.

For more information about multipart uploads, see [Multipart Upload Overview](#) in the *Amazon Simple Storage Service Developer Guide*.

If you have configured a lifecycle rule to abort incomplete multipart uploads, the upload must complete within the number of days specified in the bucket lifecycle configuration. Otherwise, the incomplete multipart upload becomes eligible for an abort operation and Amazon S3 aborts the multipart upload. For more information, see [Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy](#) in the *Amazon Simple Storage Service Developer Guide*.

For information about the permissions required to use the multipart upload API, see [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.

For request signing, multipart upload is just a series of regular requests. You initiate a multipart upload, send one or more requests to upload parts, and then complete the multipart upload process. You sign each request individually. There is nothing special about signing multipart upload requests. For more information about signing, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

### Note

After you initiate a multipart upload and upload one or more parts, to stop being charged for storing the uploaded parts, you must either complete or abort the multipart upload. Amazon S3 frees up the space used to store the parts and stop charging you for storing them only *after* you either complete or abort a multipart upload.

You can optionally request server-side encryption. For server-side encryption, Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. You can provide your own encryption key, or use AWS Key Management Service (AWS KMS) encryption keys or Amazon S3-managed encryption keys. If you choose to provide your own encryption key, the request headers you provide in [Upload Part \(p. 508\)](#) and [Upload Part - Copy \(p. 514\)](#) requests must match the headers you used in the request to initiate the upload by using [Initiate Multipart Upload \(p. 492\)](#). For more information, see [Protecting Data Using Server-Side Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
POST /ObjectName?uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This operation does not use request parameters.

## Request Headers

Name	Description	Required
Cache-Control	<p>Can be used to specify caching behavior along the request/reply chain. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
Content-Disposition	<p>Specifies presentational information for the object. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1">http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
Content-Encoding	<p>Specifies the content encodings that have been applied to the object and which decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
Content-Type	<p>A standard MIME type that describes the format of the object data. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17</a>.</p> <p>Type: String</p> <p>Default: <code>binary/octet-stream</code></p> <p>Constraints: MIME types only</p>	No
Expires	<p>The date and time at which the object should no longer be cached. For more information, see <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
x-amz-meta-	<p>Headers starting with this prefix are user-defined metadata. Each one is stored and returned as a set of key-value pairs. Amazon S3 doesn't validate or interpret user-defined metadata. For more information, see <a href="#">PUT Object (p. 412)</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
x-amz-storage-class	<p>The type of storage to use for the object that is created after a successful multipart upload. If you don't specify a class, Amazon S3 uses the default storage class, Standard. Amazon S3 supports</p>	No

Name	Description	Required
	<p>other storage classes. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Enum</p> <p>Default: STANDARD</p> <p>Valid Values: STANDARD   STANDARD_IA   ONEZONE_IA   INTELLIGENT_TIERING   GLACIER   REDUCED_REDUNDANCY</p>	
x-amz-tagging	<p>Specifies a set of one or more tags you want associated with the object. These tags are stored in the tagging subresource associated with the object.</p> <p>For more information about adding tags to an object, see <a href="#">Object Tagging Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The encoding for tags will be URL query parameter encoding. The maximum size of this header is limited to 2 K.</p>	No
x-amz-website-redirect-location	<p>If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata. For information about object metadata, see <a href="#">Object Key and Metadata</a>.</p> <p>In the following example, the request header sets the redirect to an object (<code>anotherPage.html</code>) in the same bucket:</p> <pre>x-amz-website-redirect-location: /anotherPage.html</pre> <p>In the following example, the request header sets the object redirect to another website:</p> <pre>x-amz-website-redirect-location: http://www.example.com/</pre> <p>For more information about website hosting in Amazon S3, see <a href="#">Hosting Websites on Amazon S3</a> and <a href="#">How to Configure Website Page Redirects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The value must be prefixed by, "/", "http://" or "https://". The length of the value is limited to 2 K.</p>	No

## Access Control List (ACL) Specific Request Headers

Additionally, you can use the following access control-related headers with this operation. By default, all objects are private and only the owner has full access control. When adding a new object, you can grant permissions to individual AWS accounts or predefined groups defined by Amazon S3. These permissions are then added to the Access Control List (ACL) on the object. For more information, see [Access Control List \(ACL\) Overview](#) in the *Amazon Simple Storage Service Developer Guide*. This operation enables you to grant access permissions using one of the following methods:

- **Specify canned ACL** – Amazon S3 supports a set of predefined ACLs, known as canned ACLs. Each canned ACL has a predefined set of grantees and permissions. For more information, see [Canned ACL](#).

Name	Description	Required
x-amz-acl	<p>The canned ACL to apply to the object.</p> <p>Type: String</p> <p>Default: private</p> <p>Valid Values: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control</p> <p>Constraints: None</p>	No

- **Specify access permissions explicitly** – If you want to explicitly grant access permissions to specific AWS accounts or groups, use the following headers. Each of these headers maps to specific permissions that Amazon S3 supports in an access control list (ACL). For more information, see [Access Control List \(ACL\) Overview](#). In the header, you specify a list of grantees who get the specific permission.

Name	Description	Required
x-amz-grant-read	<p>Allows the grantee to read the object data and its metadata.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-write	<p>Not applicable.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-read-acp	<p>Allows the grantee to read the object ACL.</p> <p>Type: String</p> <p>Default: None</p>	No

Name	Description	Required
	Constraints: None	
x-amz-grant-write-acp	<p>Allows the grantee to write the ACL for the applicable object.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No
x-amz-grant-full-control	<p>Grants the grantee the READ, READ_ACP, and WRITE_ACP permissions on the object.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: None</p>	No

You specify each grantee as a `type=value` pair, where the type can be one of the following:

- **emailAddress** – If the specified value is the email address of an AWS account.
- **id** – If the specified value is the canonical user ID of an AWS account.
- **uri** – If you are granting permission to a predefined group.

For example, the following `x-amz-grant-read` header grants read object data and its metadata permissions to the AWS accounts identified by their email addresses:

```
x-amz-grant-read: emailAddress="xyz@amazon.com", emailAddress="abc@amazon.com"
```

## Server-Side Encryption–Specific Request Headers

You can optionally tell Amazon S3 to encrypt data at rest using server-side encryption. Server-side encryption is for data encryption at rest. Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it when you access it. Depending on whether you want to use AWS-managed encryption keys or provide your own encryption keys, you use the following headers:

- Use encryption keys managed by AWS KMS or Amazon S3 – If you want AWS to manage the keys used to encrypt data, specify the following headers in the request.

Name	Description	Required
x-amz-server-side-encryption	<p>Specifies a server-side encryption algorithm to use when Amazon S3 creates an object.</p> <p>Type: String</p> <p>Valid Value: <code>aws:kms</code>, <code>AES256</code></p>	Yes
x-amz-server-side-encryption-aws-kms-key-id	If the <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code> , this header specifies the ID of the AWS	Yes, if the value of <code>x-amz-</code>

Name	Description	Required
	<p>Key Management Service (AWS KMS) master encryption key that was used for the object.</p> <p>Type: String</p>	server-side-encryption is aws:kms
x-amz-server-side-encryption-context	<p>If x-amz-server-side-encryption is present, and if its value is aws:kms, this header specifies the encryption context for the object. The value of this header is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs.</p> <p>Type: String</p>	No

**Note**

If you specify x-amz-server-side-encryption:aws:kms, but do not provide x-amz-server-side-encryption-aws-kms-key-id, Amazon S3 uses the default AWS KMS key to protect the data.

For more information on Server-Side Encryption with Amazon KMS-Managed Keys (SSE-KMS), see [Protecting Data Using Server-Side Encryption with AWS KMS-Managed Keys](#) in the *Amazon Simple Storage Service Developer Guide*.

- Use customer-provided encryption keys – If you want to manage your own encryption keys, provide all the following headers in the request.

Name	Description	Required
x-amz-server-side-encryption-customer-algorithm	<p>Specifies the algorithm to use to when encrypting the object.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Value: AES256</p> <p>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-key and x-amz-server-side-encryption-customer-key-MD5 headers</p>	Yes
x-amz-server-side-encryption-customer-key	<p>Specifies the customer-provided base64-encoded encryption key that Amazon S3 uses in encrypting data. This value stores the object and then is discarded. Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the x-amz-server-side-encryption-customer-algorithm header.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key-MD5 headers</p>	Yes

Name	Description	Required
x-amz-server-side-encryption-customer-key-MD5	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header for a message integrity check to ensure that the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid x-amz-server-side-encryption-customer-algorithm and x-amz-server-side-encryption-customer-key headers</p>	Yes

For more information on Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C), see [Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys \(SSE-C\)](#) in the *Amazon Simple Storage Service Developer Guide*.

## Request Elements

This operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
x-amz-abort-date	<p>If the bucket has a lifecycle rule configured with an action to abort incomplete multipart uploads and the prefix in the lifecycle rule matches the object name in the request, the response includes this header. The header indicates when the initiated multipart upload becomes eligible for an abort operation. For more information, see <a href="#">Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The response also includes the x-amz-abort-rule-id header that provides the ID of the lifecycle configuration rule that defines this action.</p> <p>Type: String</p>
x-amz-abort-rule-id	<p>This header is returned along with the x-amz-abort-date header. It identifies the applicable lifecycle configuration rule that defines the action to abort incomplete multipart uploads.</p> <p>Type: String</p>
x-amz-server-side-encryption	<p>If you specified server-side encryption either with an AWS KMS key or an Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the part that you uploaded.</p>

Name	Description
	Type: String
x-amz-server-side-encryption-aws-kms-key-id	If <code>x-amz-server-side-encryption</code> is present and has the value of <code>aws:kms</code> , this header specifies the ID of the AWS KMS master encryption key that was used for the object.  Type: String
x-amz-server-side-encryption-customer-algorithm	If server-side encryption with customer-provided encryption keys was requested, the response includes this header to confirm which encryption algorithm was used.  Type: String  Valid Values: AES256
x-amz-server-side-encryption-customer-key-MD5	If server-side encryption using a customer-provided encryption key was requested, the response returns this header to verify the integrity of the roundtrip message of the customer-provided encryption key.  Type: String

## Response Elements

Name	Description
InitiateMultipartUploadResult	Container for the response.  Type: Container  Children: Bucket, Key, UploadId  Ancestors: None
Bucket	Name of the bucket to which the multipart upload was initiated.  Type: String  Ancestors: InitiateMultipartUploadResult
Key	Object key for which the multipart upload was initiated.  Type: String  Ancestors: InitiateMultipartUploadResult
UploadId	ID for the initiated multipart upload.  Type: String  Ancestors: InitiateMultipartUploadResult

## Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

This operation initiates a multipart upload for the `example-object` object.

```
POST /example-object?uploads HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPFtaOfg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 197
Connection: keep-alive
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>VXBsb2FkIE1EIGZvciA2aWWpbmcnycBteS1tb3ZpZS5tMnRzIHVwbG9hZA</UploadId>
</InitiateMultipartUploadResult>
```

## Sample: Initiate a Multipart Upload Using Server-side Encryption with Customer-provided Encryption Keys

This example, which initiates a multipart upload request, specifies server-side encryption with customer-provided encryption keys by adding relevant headers.

```
POST /example-object?uploads HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Authorization: authorization string
Date: Wed, 28 May 2014 19:34:57 +0000
x-amz-server-side-encryption-customer-key: g01CfA3Dv40jzz5SQJ1ZukLRFqtI5WorC/8SEEXAMPLE
x-amz-server-side-encryption-customer-key-MD5: ZjOrne1X/iTcskbY2example
x-amz-server-side-encryption-customer-algorithm: AES256
```

In the response, Amazon S3 returns an `UploadId`. In addition, Amazon S3 returns the encryption algorithm and the MD5 digest of the encryption key that you provided in the request.

```
HTTP/1.1 200 OK
x-amz-id-2: 36HRCaIGp57F1FvWvVRrvd3hNn9WoBGfEaCVHTCt8QWF00qxdHazQUgfoXAbhFWD
x-amz-request-id: 50FA1D691B62CA43
Date: Wed, 28 May 2014 19:34:58 GMT
x-amz-server-side-encryption-customer-algorithm: AES256
x-amz-server-side-encryption-customer-key-MD5: ZjOrne1X/iTcskbY2m3tFg==
```

```
Transfer-Encoding: chunked

<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult
xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>

  <UploadId>EXAMPLEJZ6e0YupT2h66iePQCc9IEbYbDUy4RTpMeoSMLPrp8Z5o1u8feSRonpvnWsKKG35tI2LB9VDPiCgTy.Gq2VxQ
</UploadId>
</InitiateMultipartUploadResult>
```

## Related Actions

- [Upload Part \(p. 508\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Parts \(p. 502\)](#)
- [List Multipart Uploads \(p. 218\)](#)

# List Parts

## Description

This operation lists the parts that have been uploaded for a specific multipart upload.

This operation must include the upload ID, which you obtain by sending the initiate multipart upload request (see [Initiate Multipart Upload \(p. 492\)](#)). This request returns a maximum of 1,000 uploaded parts. The default number of parts returned is 1,000 parts. You can restrict the number of parts returned by specifying the `max-parts` request parameter. If your multipart upload consists of more than 1,000 parts, the response returns an `IsTruncated` field with the value of `true`, and a `NextPartNumberMarker` element. In subsequent List Parts requests you can include the `part-number-marker` query string parameter and set its value to the `NextPartNumberMarker` field value from the previous response.

For more information on multipart uploads, see [Uploading Objects Using Multipart Upload](#) in the *Amazon Simple Storage Service Developer Guide*.

For information on permissions required to use the multipart upload API, see [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: authorization string
```

### Request Parameters

This implementation of GET uses the parameters in the following table to return a subset of the objects in a bucket.

Parameter	Description	Required
<code>encoding-type</code>	<p>Requests Amazon S3 to encode the response and specifies the encoding method to use.</p> <p>An object key can contain any Unicode character; however, XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid value: <code>url</code></p>	No
<code>uploadId</code>	Upload ID identifying the multipart upload whose parts are being listed. <p>Type: String</p>	Yes

Parameter	Description	Required
	Default: None	
max-parts	Sets the maximum number of parts to return in the response body.  Type: String  Default: 1,000	No
part-number-marker	Specifies the part after which listing should begin. Only parts with higher part numbers will be listed.  Type: String  Default: None	No

## Request Headers

This operation uses only Request Headers common to most requests. For more information, see [Common Request Headers \(p. 2\)](#).

## Request Elements

This operation does not use request elements.

## Responses

### Response Headers

This operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

Name	Description
x-amz-abort-date	If the bucket has a lifecycle rule configured with an action to abort incomplete multipart uploads and the prefix in the lifecycle rule matches the object name in the request, then the response includes this header indicating when the initiated multipart upload will become eligible for abort operation. For more information, see <a href="#">Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy</a> in the <a href="#">Amazon Simple Storage Service Developer Guide</a> .  The response will also include the x-amz-abort-rule-id header that will provide the ID of the lifecycle configuration rule that defines this action.  Type: String
x-amz-abort-rule-id	This header is returned along with the x-amz-abort-date header. It identifies applicable lifecycle configuration rule that defines the action to abort incomplete multipart uploads.  Type: String

Name	Description
ListPartsResult	<p>Container for the response.</p> <p>Children: Bucket, Key, UploadId, Initiator, Owner, StorageClass, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part</p> <p>Type: Container</p>
Bucket	<p>Name of the bucket to which the multipart upload was initiated.</p> <p>Type: String</p> <p>Ancestor: ListPartsResult</p>
Encoding-Type	<p>Encoding type used by Amazon S3 to encode object key names in the XML response.</p> <p>If you specify encoding-type request parameter, Amazon S3 includes this element in the response, and returns encoded key name values in the Key element.</p> <p>Type: String</p> <p>Ancestor: ListBucketResult</p>
Key	<p>Object key for which the multipart upload was initiated.</p> <p>Type: String</p> <p>Ancestor: ListPartsResult</p>
UploadId	<p>Upload ID identifying the multipart upload whose parts are being listed.</p> <p>Type: String</p> <p>Ancestor: ListPartsResult</p>
Initiator	<p>Container element that identifies who initiated the multipart upload. If the initiator is an AWS account, this element provides the same information as the Owner element. If the initiator is an IAM User, then this element provides the user ARN and display name.</p> <p>Children: ID, DisplayName</p> <p>Type: Container</p> <p>Ancestor: ListPartsResult</p>
ID	<p>If the principal is an AWS account, it provides the Canonical User ID. If the principal is an IAM User, it provides a user ARN value.</p> <p>Type: String</p> <p>Ancestor: Initiator</p>

Name	Description
DisplayName	<p>Principal's name.</p> <p>Type: String</p> <p>Ancestor: Initiator</p>
Owner	<p>Container element that identifies the object owner, after the object is created. If multipart upload is initiated by an IAM user, this element provides the parent account ID and display name.</p> <p>Children: ID, DisplayName</p> <p>Type: Container</p> <p>Ancestor: ListPartsResult</p>
StorageClass	<p>Class of storage (STANDARD or REDUCED_REDUNDANCY) used to store the uploaded object.</p> <p>Type: String</p> <p>Ancestor: ListPartsResult</p>
PartNumberMarker	<p>Part number after which listing begins.</p> <p>Type: Integer</p> <p>Ancestor: ListPartsResult</p>
NextPartNumberMarker	<p>When a list is truncated, this element specifies the last part in the list, as well as the value to use for the <code>part-number-marker</code> request parameter in a subsequent request.</p> <p>Type: Integer</p> <p>Ancestor: ListPartsResult</p>
MaxParts	<p>Maximum number of parts that were allowed in the response.</p> <p>Type: Integer</p> <p>Ancestor: ListPartsResult</p>
IsTruncated	<p>Indicates whether the returned list of parts is truncated. A <code>true</code> value indicates that the list was truncated. A list can be truncated if the number of parts exceeds the limit returned in the <code>MaxParts</code> element.</p> <p>Type: Boolean</p> <p>Ancestor: ListPartsResult</p>
Part	<p>Container for elements related to a particular part. A response can contain zero or more <code>Part</code> elements.</p> <p>Children: PartNumber, LastModified, ETag, Size</p> <p>Type: String</p> <p>Ancestor: ListPartsResult</p>

Name	Description
PartNumber	Part number identifying the part.  Type: Integer  Ancestor: Part
LastModified	Date and time at which the part was uploaded.  Type: Date  Ancestor: Part
ETag	Entity tag returned when the part was uploaded.  Type: String  Ancestor: Part
Size	Size in bytes of the uploaded part data.  Type: Integer  Ancestor: Part

## Examples

### Sample Request

Assume you have uploaded parts with sequential part numbers starting with 1. The following List Parts request specifies `max-parts` and `part-number-marker` query parameters. The request lists the first two parts that follow part number 1, that is, you will get parts 2 and 3 in the response. If more parts exist, the result is a truncated result and therefore the response will return an `IsTruncated` element with the value `true`. The response will also return the `NextPartNumberMarker` element with the value 3, which should be used for the value of the `part-number-marker` request query string parameter in the next List Parts request.

```
GET /example-object?
uploadId=XXBsb2FkIE1EIGZvcIBlbHZpbmcncyVcdS1tb3ZpZS5tMnRzEEEwbG9hZA&max-parts=2&part-
number-marker=1 HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: authorization string
```

### Sample Response

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 985
Connection: keep-alive
Server: AmazonS3
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListPartsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Bucket>example-bucket</Bucket>
  <Key>example-object</Key>
  <UploadId>XXBsb2FkIElEIGZvciBlbHZpbmcncyVcdS1tb3ZpZS5tMnRzEEEwbG9hZA</UploadId>
  <Initiator>
    <ID>arn:aws:iam::111122223333:user/some-user-11116a31-17b5-4fb7-9df5-b288870f11xx</ID>
    <DisplayName>umat-user-11116a31-17b5-4fb7-9df5-b288870f11xx</DisplayName>
  </Initiator>
  <Owner>
    <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeef76c078efc7c6caea54ba06a</ID>
    <DisplayName>someName</DisplayName>
  </Owner>
  <StorageClass>STANDARD</StorageClass>
  <PartNumberMarker>1</PartNumberMarker>
  <NextPartNumberMarker>3</NextPartNumberMarker>
  <MaxParts>2</MaxParts>
  <IsTruncated>true</IsTruncated>
  <Part>
    <PartNumber>2</PartNumber>
    <LastModified>2010-11-10T20:48:34.000Z</LastModified>
    <ETag>"7778aef83f66abc1fa1e8477f296d394"</ETag>
    <Size>10485760</Size>
  </Part>
  <Part>
    <PartNumber>3</PartNumber>
    <LastModified>2010-11-10T20:48:33.000Z</LastModified>
    <ETag>"aaaa18db4cc2f85cedef654fcc4a4x8"</ETag>
    <Size>10485760</Size>
  </Part>
</ListPartsResult>
```

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)
- [Upload Part \(p. 508\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Multipart Uploads \(p. 218\)](#)

# Upload Part

## Description

This operation uploads a part in a multipart upload.

### Note

In this operation, you provide part data in your request. However, you have an option to specify your existing Amazon S3 object as a data source for the part you are uploading. To upload a part from an existing object, you use the Upload Part (Copy) operation. For more information, see [Upload Part - Copy \(p. 514\)](#).

You must initiate a multipart upload (see [Initiate Multipart Upload \(p. 492\)](#)) before you can upload any part. In response to your initiate request, Amazon S3 returns an upload ID, a unique identifier, that you must include in your upload part request.

Part numbers can be any number from 1 to 10,000, inclusive. A part number uniquely identifies a part and also defines its position within the object being created. If you upload a new part using the same part number that was used with a previous part, the previously uploaded part is overwritten. Each part must be at least 5 MB in size, except the last part. There is no size limit on the last part of your multipart upload.

To ensure that data is not corrupted when traversing the network, specify the Content-MD5 header in the upload part request. Amazon S3 checks the part data against the provided MD5 value. If they do not match, Amazon S3 returns an error.

### Note

After you initiate multipart upload and upload one or more parts, you must either complete or abort multipart upload in order to stop getting charged for storage of the uploaded parts. Only after you either complete or abort the multipart upload, Amazon S3 frees up the parts storage and stops charging you for it.

For more information on multipart uploads, go to [Multipart Upload Overview](#) in the *Amazon Simple Storage Service Developer Guide*.

For information on the permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.

You can optionally request server-side encryption where Amazon S3 encrypts your data as it writes it to disks in its data centers and decrypts it for you when you access it. You have the option of providing your own encryption key, or you can use the AWS-managed encryption keys. If you choose to provide your own encryption key, the request headers you provide in the request must match the headers you used in the request to initiate the upload by using [Initiate Multipart Upload \(p. 492\)](#). For more information, go to [Using Server-Side Encryption](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Content-Length: Size
Authorization: authorization string
```

## Request Parameters

This operation does not use request parameters.

## Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
Content-Length	<p>The size of the part, in bytes. For more information, go to <a href="http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13">http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13</a>.</p> <p>Type: Integer</p> <p>Default: None</p>	Yes
Content-MD5	<p>The base64-encoded 128-bit MD5 digest of the part data. This header can be used as a message integrity check to verify that the part data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information, see <a href="#">RFC 1864</a>.</p> <p>Type: String</p> <p>Default: None</p>	No
Expect	<p>When your application uses 100-continue, it does not send the request body until it receives an acknowledgment. If the message is rejected based on the headers, the body of the message is not sent. For more information, go to <a href="#">RFC 2616</a>.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Values: 100-continue</p>	No

## Server-Side Encryption Specific Request Headers

Server-side encryption is supported by the S3 Multipart Upload actions. Unless you are using a customer-provided encryption key, you don't need to specify the encryption parameters in each UploadPart request. Instead, you only need to specify the server side encryption parameters in the initial Initiate Multipart request. For more information, see [Initiate Multipart Upload \(p. 492\)](#).

If you requested server-side encryption using a customer-provided encryption key in your initiate multipart upload request, you must provide identical encryption information in each part upload using the following headers.

Name	Description	Required
x-amz-server-side-encryption	Specifies the algorithm to use to when encrypting the object.	Yes

Name	Description	Required
-customer-algorithm	<p>Type: String</p> <p>Default: None</p> <p>Valid Value: AES256</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	
<code>x-amz-server-side-encryption-customer-key</code>	<p>Specifies the customer-provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This value is used to store the object and then is discarded; Amazon does not store the encryption key. The key must be appropriate for use with the algorithm specified in the <code>x-amz-server-side-encryption-customer-algorithm</code> header.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
<code>x-amz-server-side-encryption-customer-key-MD5</code>	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</p>	Yes

## Request Elements

This operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation can include the following response headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
<code>x-amz-server-side-encryption</code>	If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.

Name	Description
	Type: String
x-amz-server-side-encryption-aws-kms-key-id	If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.  Type: String
x-amz-server-side-encryption-customer-algorithm	If server-side encryption with customer-provided encryption keys(SSE-C) encryption was requested, the response will include this header confirming the encryption algorithm used.  Type: String  Valid Values: AES256
x-amz-server-side-encryption-customer-key-MD5	If SSE-C encryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key.  Type: String
x-amz-storage-class	Provides storage class information of the object. Amazon S3 returns this header for all objects except for Standard storage class objects.  For more information, go to <a href="#">Storage Classes in Amazon Simple Storage Service Developer Guide</a> .  Type: String  Default: None

## Response Elements

This operation does not use response elements.

## Special Errors

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
NoSuchUpload	The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.	404 Not Found	Client

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Sample Request

The following PUT request uploads a part (part number 1) in a multipart upload. The request includes the upload ID that you get in response to your Initiate Multipart Upload request.

```
PUT /my-movie.m2ts?  
partNumber=1&uploadId=VCVsb2FkIE1EIGZvcIBlbZZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1  
Host: example-bucket.s3.amazonaws.com  
Date: Mon, 1 Nov 2010 20:34:56 GMT  
Content-Length: 10485760  
Content-MD5: pUNXr/BjKK5G2UKvaRRrOA==  
Authorization: authorization string  
  
***part data omitted***
```

## Sample Response

The response includes the ETag header. You need to retain this value for use when you send the Complete Multipart Upload request.

```
HTTP/1.1 200 OK  
x-amz-id-2: Vvag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPFtaOFg==  
x-amz-request-id: 656c76696e6727732072657175657374  
Date: Mon, 1 Nov 2010 20:34:56 GMT  
ETag: "b54357faf0632cce46e942fa68356b38"  
Content-Length: 0  
Connection: keep-alive  
Server: AmazonS3
```

## Sample: Upload a part with an encryption key in the request for server-side encryption

If you initiated a multipart upload, see [Sample: Initiate a Multipart Upload Using Server-side Encryption with Customer-provided Encryption Keys \(p. 500\)](#), with a request to save an object using server-side encryption with a customer-provided encryption key, each part upload must also include the same set of encryption-specific headers as shown in the following example request.

```
PUT /example-object?  
partNumber=1&uploadId=EXAMPLEJZ6e0YupT2h66iePQCc9IEbYbDUy4RTpMeoSMLPRp8Z5o1u8feSRonpvnWsKKG35ti2LB9VDPi  
HTTP/1.1  
Host: example-bucket.s3.amazonaws.com  
Authorization: authorization string  
Date: Wed, 28 May 2014 19:40:11 +0000  
x-amz-server-side-encryption-customer-key: g01CfA3Dv40jZz5SQJ1ZukLRFqtI5WorC/8SEEXAMPLE  
x-amz-server-side-encryption-customer-key-MD5: ZjOrne1X/iTcskbY2example  
x-amz-server-side-encryption-customer-algorithm: AES256
```

In the response, Amazon S3 returns encryption-specific headers providing the encryption algorithm used and MD5 digest of the encryption key you provided in the request.

```
HTTP/1.1 100 Continue HTTP/1.1 200 OK  
x-amz-id-2: Zn8bf8aEFQ+kBnGPBc/JaAf9SoWM68QDPS9+SyFwkIZOHUG2BiRLZi5oXw4cOCET  
x-amz-request-id: 5A37448A37622243  
Date: Wed, 28 May 2014 19:40:12 GMT  
ETag: "7e10e7d25dc4581d89b9285be5f384fd"  
x-amz-server-side-encryption-customer-algorithm: AES256  
x-amz-server-side-encryption-customer-key-MD5: ZjOrne1X/iTcskbY2example
```

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)

- [Complete Multipart Upload \(p. 486\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Parts \(p. 502\)](#)
- [List Multipart Uploads \(p. 218\)](#)

# Upload Part - Copy

## Description

Uploads a part by copying data from an existing object as data source. You specify the data source by adding the request header `x-amz-copy-source` in your request and a byte range by adding the request header `x-amz-copy-source-range` in your request.

The minimum allowable part size for a multipart upload is 5 MB. For more information about multipart upload limits, go to [Quick Facts](#) in the *Amazon Simple Storage Service Developer Guide*.

### Note

Instead of using an existing object as part data, you might use the [Upload Part](#) operation and provide data in your request. For more information, see [Upload Part \(p. 508\)](#).

You must initiate a multipart upload before you can upload any part. In response to your initiate request, Amazon S3 returns a unique identifier, the upload ID, that you must include in your upload part request.

**For more information on using the upload part - copy operation, see the following topics:**

- For conceptual information on multipart uploads, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon Simple Storage Service Developer Guide*.
- For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon Simple Storage Service Developer Guide*.
- For information about copying objects using a single atomic operation vs. the multipart upload, go to [Operations on Objects](#) in the *Amazon Simple Storage Service Developer Guide*.
- For information about using server-side encryption with customer-provided encryption keys with the upload part - copy operation, see [PUT Object - Copy \(p. 431\)](#) and [Upload Part \(p. 508\)](#).

## Requests

### Syntax

```
PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
x-amz-copy-source: /source_bucket/sourceObject
x-amz-copy-source-range:bytes=first-last
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
Date: date
Authorization: authorization string
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

This implementation of the operation can use the following request headers in addition to the request headers common to all operations. Request headers are limited to 8 KB in size. For more information, see [Common Request Headers \(p. 2\)](#).

Name	Description	Required
x-amz-copy-source	<p>The name of the source bucket and the source object key name separated by a slash ('/').</p> <p>Type: String</p> <p>Default: None</p>	Yes
x-amz-copy-source-range	<p>The range of bytes to copy from the source object. The range value must use the form <code>bytes=first-last</code>, where the first and last are the zero-based byte offsets to copy. For example, <code>bytes=0-9</code> indicates that you want to copy the first ten bytes of the source.</p> <p>This request header is not required when copying an entire source object.</p> <p>Type: Integer</p> <p>Default: None</p>	No

The following conditional headers are based on the object that the `x-amz-copy-source` header specifies.

Name	Description	Required
x-amz-copy-source-if-match	<p>Perform a copy if the source object entity tag (ETag) matches the specified value. If the value does not match, Amazon S3 returns an HTTP status code <i>412 precondition failed</i> error.</p> <p>See Consideration 1 after the table.</p> <p>Type: String</p> <p>Default: None</p>	No
x-amz-copy-source-if-none-match	<p>Perform a copy if the source object entity tag (ETag) is different than the value specified using this header. If the values match, Amazon S3 returns an HTTP status code <i>412 precondition failed</i> error.</p> <p>See Consideration 2 after the table.</p> <p>Type: String</p> <p>Default: None</p>	No
x-amz-copy-source-if-unmodified-since	<p>Perform a copy if the source object is not modified after the time specified using this header. If the source object is modified, Amazon S3 returns an HTTP status code <i>412 precondition failed</i> error.</p> <p>See Consideration 1 after the table.</p> <p>Type: String</p>	No

Name	Description	Required
	Default: None	
x-amz-copy-source-if-modified-since	<p>Perform a copy if the source object is modified after the time specified using this header. If the source object is not modified, Amazon S3 returns an HTTP status code <i>412 precondition failed</i> error.</p> <p>See Consideration 2 after the table.</p> <p>Type: String</p> <p>Default: None</p>	No

Note the following additional considerations about the preceding request headers:

- **Consideration 1** – If both of the `x-amz-copy-source-if-match` and `x-amz-copy-source-if-unmodified-since` headers are present in the request as follows:
 

`x-amz-copy-source-if-match` condition evaluates to `true`, and;

`x-amz-copy-source-if-unmodified-since` condition evaluates to `false`;

then, S3 returns `200 OK` and copies the data.
- **Consideration 2** – If both of the `x-amz-copy-source-if-none-match` and `x-amz-copy-source-if-modified-since` headers are present in the request as follows:
 

`x-amz-copy-source-if-none-match` condition evaluates to `false`, and;

`x-amz-copy-source-if-modified-since` condition evaluates to `true`;

then, S3 returns `412 Precondition Failed` response code.

## Server-Side Encryption Specific Request Headers

If you requested server-side encryption using a customer-provided encryption key in your initiate multipart upload request, you must provide identical encryption information in each part upload using the following headers.

Name	Description	Required
x-amz-server-side-encryption-customer-algorithm	<p>Specifies the algorithm to use to when encrypting the object.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Value: <code>AES256</code></p> <p>Constraints: Must be accompanied by a valid <code>x-amz-server-side-encryption-customer-key</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes

Name	Description	Required
x-amz-server-side-encryption-customer-key	<p>Specifies the customer provided base64-encoded encryption key for Amazon S3 to use in encrypting data. This must be the same encryption key specified in the initiate multipart upload request.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by a valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
x-amz-server-side-encryption-customer-key-MD5	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header as a message integrity check to ensure the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by a valid <code>x-amz-server-side-encryption-customer-algorithm</code> and <code>x-amz-server-side-encryption-customer-key</code> headers.</p>	Yes

If the source object is encrypted using server-side encryption with a customer-provided encryption key, you must use the following headers providing encryption information so that Amazon S3 can decrypt the object for copying.

Name	Description	Required
x-amz-copy-source-server-side-encryption-customer-algorithm	<p>Specifies algorithm to use when decrypting the source object.</p> <p>Type: String</p> <p>Default: None</p> <p>Valid Value: AES256</p> <p>Constraints: Must be accompanied by a valid <code>x-amz-copy-source-server-side-encryption-customer-key</code> and <code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code> headers.</p>	Yes
x-amz-copy-source-server-side-encryption-customer-key	<p>Specifies the customer provided base-64 encoded encryption key for Amazon S3 to use to decrypt the source object. The encryption key provided in this header must be one that was used when the source object was created.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by a valid <code>x-amz-copy-source-server-side-encryption-customer-algorithm</code></p>	Yes

Name	Description	Required
	and <code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code> headers.	
<code>x-amz-copy-source-server-side-encryption-customer-key-MD5</code>	<p>Specifies the base64-encoded 128-bit MD5 digest of the encryption key according to <a href="#">RFC 1321</a>. Amazon S3 uses this header for a message integrity check to ensure the encryption key was transmitted without error.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: Must be accompanied by a valid <code>x-amz-copy-source-server-side-encryption-customer-algorithm</code> and <code>x-amz-copy-source-server-side-encryption-customer-key</code> headers.</p>	Yes

## Request Elements

This operation does not use request elements.

## Versioning

If your bucket has versioning enabled, you could have multiple versions of the same object. By default, `x-amz-copy-source` identifies the current version of the object to copy. If the current version is a delete marker and you don't specify a `versionId` in the `x-amz-copy-source`, Amazon S3 returns a 404 error, because the object does not exist. If you specify `versionId` in the `x-amz-copy-source` and the `versionId` is a delete marker, Amazon S3 returns an HTTP 400 error, because you are not allowed to specify a delete marker as a version for the `x-amz-copy-source`.

You can optionally specify a specific version of the source object to copy by adding the `versionId` subresource as shown in the following example:

```
x-amz-copy-source: /bucket/object?versionId=version id
```

## Responses

### Response Headers

This implementation of the operation can include the following headers in addition to the response headers common to all responses. For more information, see [Common Response Headers \(p. 4\)](#).

Name	Description
<code>x-amz-copy-source-version-id</code>	The version of the source object that was copied, if you have enabled versioning on the source bucket.  Type: String
<code>x-amz-server-side-encryption</code>	If you specified server-side encryption either with an AWS KMS or Amazon S3-managed encryption key in your initiate multipart

Name	Description
	upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the object.  Type: String
x-amz-server-side-encryption-aws-kms-key-id	If the x-amz-server-side-encryption is present and has the value of aws:kms, this header specifies the ID of the AWS Key Management Service (KMS) master encryption key that was used for the object.  Type: String
x-amz-server-side-encryption-customer-algorithm	If server-side encryption with customer-provided encryption keys encryption was requested, the response will include this header confirming the encryption algorithm used.  Type: String  Valid Values: AES256
x-amz-server-side-encryption-customer-key-MD5	If server-side encryption with customer-provided encryption keys encryption was requested, the response includes this header to provide roundtrip message integrity verification of the customer-provided encryption key.  Type: String

## Response Elements

Name	Description
CopyPartResult	Container for all response elements.  Type: Container  Ancestor: None
ETag	Returns the ETag of the new part.  Type: String  Ancestor: CopyPartResult
LastModified	Returns the date the part was last modified.  Type: String  Ancestor: CopyPartResult

### Important

Part boundaries are factored into ETag calculations, so if the part boundary on the source is different than on the destination, then the ETag data will not match between the two. However, data integrity checks are performed with each copy to ensure that the data written to the destination matches the data at the source.

## Special Errors

Error Code	Description	HTTP Status Code
NoSuchUpload	The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed.	404 Not Found
InvalidRequest	The specified copy source is not supported as a byte-range copy source.	400 Bad Request

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

As the following examples illustrate, when a request succeeds, Amazon S3 returns <CopyPartResult> in the body. If you included `versionId` in the request, Amazon S3 returns the version ID in the `x-amz-copy-source-version-id` response header.

### Sample Request

The following `PUT` request uploads a part (part number 2) in a multipart upload. The request specifies a byte range from an existing object as the source of this upload. The request includes the upload ID that you get in response to your `Initiate Multipart Upload` request.

```
PUT /newobject?
partNumber=2&uploadId=VCVsb2FkIE1EIGZvcIBlbZZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1
Host: target-bucket.s3.amazonaws.com
Date: Mon, 11 Apr 2011 20:34:56 GMT
x-amz-copy-source: /source-bucket/sourceobject
x-amz-copy-source-range:bytes=500-6291456
Authorization: authorization string
```

### Sample Response

The response includes the `ETag` value. You need to retain this value to use when you send the `Complete Multipart Upload` request.

```
HTTP/1.1 200 OK
x-amz-id-2: Vvag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==
x-amz-request-id: 656c76696e6727732072657175657374
Date: Mon, 11 Apr 2011 20:34:56 GMT
Server: AmazonS3

<CopyPartResult>
  <LastModified>2011-04-11T20:34:56.000Z</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>
```

### Sample Request

The following `PUT` request uploads a part (part number 2) in a multipart upload. The request does not specify the optional byte range header, but requests the entire source object copy as part 2. The request includes the upload ID that you got in response to your `Initiate Multipart Upload` request.

```
PUT /newobject?  
partNumber=2&uploadId=VCVsb2FkIE1EIGZvcIBlbZZpbmcnycBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1  
Host: target-bucket.s3.amazonaws.com  
Date: Mon, 11 Apr 2011 20:34:56 GMT  
x-amz-copy-source: /source-bucket/sourceobject  
Authorization: authorization string  
Sample Response
```

The response structure is similar to the one specified in the preceding example.

## Sample Request

The following PUT request uploads a part (part number 2) in a multipart upload. The request specifies a specific version of the source object to copy by adding the `versionId` subresource. The byte range requests 6 MB of data, starting with byte 500, as the part to be uploaded.

```
PUT /newobject?  
partNumber=2&uploadId=VCVsb2FkIE1EIGZvcIBlbZZpbmcnycBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1  
Host: target-bucket.s3.amazonaws.com  
Date: Mon, 11 Apr 2011 20:34:56 GMT  
x-amz-copy-source: /source-bucket/sourceobject?versionId=3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY  
+MTRCx3vJBH4ONr8X8gdRQBpUMLUo  
x-amz-copy-source-range:bytes=500-6291456  
Authorization: authorization string
```

## Sample Response

The response includes the ETag value. You need to retain this value to use when you send the Complete Multipart Upload request.

```
HTTP/1.1 200 OK  
x-amz-id-2: Vvag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/C1NPcfTWAtRPfTaOFg==  
x-amz-request-id: 656c76696e6727732072657175657374  
x-amz-copy-source-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY  
+MTRCx3vJBH4ONr8X8gdRQBpUMLUo  
Date: Mon, 11 Apr 2011 20:34:56 GMT  
Server: AmazonS3  
  
<CopyPartResult>  
  <LastModified>2011-04-11T20:34:56.000Z</LastModified>  
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>  
</CopyPartResult>
```

## Related Actions

- [Initiate Multipart Upload \(p. 492\)](#)
- [Upload Part \(p. 508\)](#)
- [Complete Multipart Upload \(p. 486\)](#)
- [Abort Multipart Upload \(p. 484\)](#)
- [List Parts \(p. 502\)](#)
- [List Multipart Uploads \(p. 218\)](#)

# ObjectLockLegalHold

Service: Amazon Simple Storage Service

A Legal Hold configuration for an object.

## Contents

### Status

Indicates whether the specified object has a Legal Hold in place.

Type: String

Valid Values: ON | OFF

Required: Yes

# ObjectLockRetention

Service: Amazon Simple Storage Service

A Retention configuration for an object.

## Contents

### Mode

Indicates the Retention mode for the specified object.

Type: String

Valid Values: GOVERNANCE | COMPLIANCE

Required: Yes

### RetainUntilDate

Type: Timestamp

Format: *2020-01-05T00:00:00.000Z*

Required: Yes

# Amazon S3 Resources

Following is a table that lists related resources that you'll find useful as you work with this service.

Resource	Description
<a href="#">Amazon Simple Storage Service Getting Started Guide</a>	The getting started guide provides a quick tutorial of the service based on a simple use case.
<a href="#">Amazon Simple Storage Service Developer Guide</a>	The developer guide describes how to accomplish tasks using Amazon S3 operations.
<a href="#">Amazon S3 Technical FAQ</a>	The FAQ covers the top 20 questions developers have asked about this product.
<a href="#">Amazon S3 Release Notes</a>	The Release Notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
<a href="#">Tools for Amazon Web Services</a>	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS SDKs and tools.
<a href="#">AWS Management Console</a>	The console allows you to perform most of the functions of Amazon S3 without programming.
<a href="#">Discussion Forums</a>	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
<a href="#">AWS Support Center</a>	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.
<a href="#">AWS Premium Support</a>	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
<a href="#">Amazon S3 product information</a>	The primary web page for information about Amazon S3.
<a href="#">Contact Us</a>	A central contact point for inquiries concerning AWS billing, account, events, abuse, etc.
<a href="#">Conditions of Use</a>	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

# Document History

The following table describes the important changes to the documentation since the last release of the *Amazon Simple Storage Service API Reference*.

- **API version:** 2006-03-01
- **Latest documentation update:** December 04, 2018

Change	Description	Release Date
Support for Parquet-formatted Amazon S3 inventory files	<p>Amazon S3 now supports the <a href="#">Apache Parquet (Parquet)</a> format in addition to the <a href="#">Apache optimized row columnar (ORC)</a> and comma-separated values (CSV) file formats for inventory output files. For more information, see <a href="#">Amazon S3 Inventory</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs were updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">GET Bucket Inventory (p. 139)</a></li> <li>• <a href="#">PUT Bucket inventory (p. 258)</a></li> </ul>	December 04, 2018
PUT directly to the GLACIER storage class	<p>The Amazon S3 PUT and related operations now support specifying GLACIER as the storage class when creating objects. Previously, you had to transition to the GLACIER storage class from another Amazon S3 storage class. For more information about the GLACIER storage class, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs were updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Object (p. 412)</a></li> <li>• <a href="#">POST Object (p. 385)</a></li> <li>• <a href="#">PUT Object - Copy (p. 431)</a></li> <li>• <a href="#">Initiate Multipart Upload (p. 492)</a></li> </ul>	November 26, 2018
Object Lock	<p>Amazon S3 now supports locking objects using a Write Once Read Many (WORM) model. You can lock objects for a definite period of time using a retention period or indefinitely using a legal hold. For more information about Amazon S3 Object Lock, see <a href="#">Locking Objects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs were updated for S3 Object Lock:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Object (p. 412)</a></li> <li>• <a href="#">GET Object (p. 349)</a></li> <li>• <a href="#">HEAD Object (p. 373)</a></li> <li>• <a href="#">PUT Bucket (p. 227)</a></li> <li>• <a href="#">HEAD Bucket (p. 204)</a></li> </ul>	November 26, 2018

Change	Description	Release Date
	<p>The following new APIs were added for S3 Object Lock:</p> <ul style="list-style-type: none"> <li>• <a href="#">GET Bucket object lock configuration (p. 169)</a></li> <li>• <a href="#">PUT Bucket object lock configuration (p. 298)</a></li> <li>• <a href="#">GET Object retention (p. 366)</a></li> <li>• <a href="#">PUT Object retention (p. 429)</a></li> <li>• <a href="#">GET Object legal hold (p. 365)</a></li> <li>• <a href="#">PUT Object legal hold (p. 427)</a></li> </ul>	
New storage class	<p>Amazon S3 now offers a new storage class named INTELLIGENT_TIERING that is for storing data that has changing or unknown access patterns. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs were updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Object (p. 412)</a></li> <li>• <a href="#">POST Object (p. 385)</a></li> <li>• <a href="#">PUT Object - Copy (p. 431)</a></li> <li>• <a href="#">Initiate Multipart Upload (p. 492)</a></li> </ul>	November 26, 2018
Block Public Access	<p>Amazon S3 now includes the ability to block public access to buckets and objects on a per-bucket or account-wide basis. For more information, see <a href="#">Using Amazon S3 Block Public Access</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following new APIs have been added:</p> <ul style="list-style-type: none"> <li>• <a href="#">GET BucketPolicyStatus (p. 170)</a></li> <li>• <a href="#">PUT PublicAccessBlock (p. 277) (Bucket)</a></li> <li>• <a href="#">GET PublicAccessBlock (p. 153) (Bucket)</a></li> <li>• <a href="#">DELETE PublicAccessBlock (p. 89) (Bucket)</a></li> <li>• <a href="#">PUT PublicAccessBlock (p. 72) (Account)</a></li> <li>• <a href="#">GET PublicAccessBlock (p. 69) (Account)</a></li> <li>• <a href="#">DELETE PublicAccessBlock (p. 68) (Account)</a></li> </ul>	November 15, 2018
Filtering enhancements in cross-region replication (CRR) rules	<p>In a CRR rule configuration, you can specify an object filter to choose a subset of objects to apply the rule to. Previously, you could filter only on an object key prefix. In this release, you can filter on an object key prefix, one or more object tags, or both. For more information, see <a href="#">Replication Configuration Overview</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs are updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Bucket replication (p. 302)</a></li> <li>• <a href="#">GET Bucket replication (p. 187)</a></li> <li>• <a href="#">DELETE Bucket replication (p. 95)</a></li> </ul>	September 19, 2018

Change	Description	Release Date
New storage class	Amazon S3 now offers a new storage class, ONEZONE_IA (IA, for infrequent access) for storing objects. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	April 4, 2018
Amazon S3 Select	<p>Amazon S3 Select is now generally available. This feature retrieves object content based on an SQL expression. For more information, see <a href="#">Selecting Content from Objects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following API has been updated:</p> <ul style="list-style-type: none"> <li>• <a href="#">SELECT Object Content (p. 457)</a></li> </ul>	April 4, 2018
Asia Pacific (Osaka-Local) Region	<p>Amazon S3 is now available in the Asia Pacific (Osaka-Local) Region. For more information about Amazon S3 Regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i>.</p> <p><b>Important</b> You can use the Asia Pacific (Osaka-Local) Region only in conjunction with the Asia Pacific (Tokyo) Region. To request access to Asia Pacific (Osaka-Local) Region, contact your sales representative.</p>	February 12, 2018
EU (Paris) Region	Amazon S3 is now available in the EU (Paris) Region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	December 18, 2017
China (Ningxia) Region	Amazon S3 is now available in the China (Ningxia) Region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	December 11, 2017
Querying archives with SQL	<p>Amazon S3 now supports querying Glacier data archives with SQL. For more information, see <a href="#">Querying Archived Objects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following API changed:</p> <ul style="list-style-type: none"> <li>• <a href="#">POST Object restore (p. 397)</a></li> </ul>	November 29, 2017
SELECT Object Content (Preview)	<p>Amazon S3 now supports the SELECT Object Content functionality as part of a Preview program. This feature retrieves object content based on an SQL expression.</p> <p>The following API has been added:</p> <ul style="list-style-type: none"> <li>• <a href="#">SELECT Object Content (p. 457)</a></li> </ul>	November 29, 2017

Change	Description	Release Date
Support for ORC-formatted Amazon S3 inventory files	<p>Amazon S3 now supports the <a href="#">Apache optimized row columnar (ORC)</a> format in addition to comma-separated values (CSV) file format for inventory output files. For more information, see <a href="#">Amazon S3 Inventory</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs are updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">GET Bucket Inventory (p. 139)</a></li> <li>• <a href="#">PUT Bucket inventory (p. 258)</a></li> </ul>	November 17, 2017
Default encryption for S3 buckets	<p>Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS). For more information, see <a href="#">Amazon S3 Default Encryption for S3 Buckets</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs are updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">DELETE Bucket encryption (p. 84)</a></li> <li>• <a href="#">GET Bucket encryption (p. 135)</a></li> <li>• <a href="#">PUT Bucket encryption (p. 254)</a></li> </ul>	November 06, 2017
Encryption status in Amazon S3 inventory	<p>Amazon S3 now supports including encryption status in Amazon S3 inventory so you can see how your objects are encrypted at rest for compliance auditing or other purposes. You can also configure to encrypt Amazon S3 inventory with server-side encryption (SSE) or SSE-KMS so that all inventory files are encrypted accordingly. For more information, see <a href="#">Amazon S3 Inventory</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following APIs are updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">GET Bucket Inventory (p. 139)</a></li> <li>• <a href="#">PUT Bucket inventory (p. 258)</a></li> </ul>	November 06, 2017

Change	Description	Release Date
Cross-region replication (CRR) enhancements	<p>Cross-region replication (CRR) now supports the following:</p> <ul style="list-style-type: none"> <li>• In a cross-account scenario, you can add a CRR configuration to change replica ownership to the AWS account that owns the destination bucket. For more information, see <a href="#">CRR: Change Replica Owner</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</li> <li>• By default, Amazon S3 does not replicate objects in your source bucket that are created using server-side encryption using AWS KMS-managed keys. In your CRR configuration, you can now direct Amazon S3 to replicate these objects. For more information, see <a href="#">CRR: Replicating Objects Created with SEE Using AWS KMS-Managed Encryption Keys</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</li> </ul> <p>The following APIs are updated accordingly:</p> <ul style="list-style-type: none"> <li>• <a href="#">GET Bucket replication (p. 187)</a></li> <li>• <a href="#">PUT Bucket replication (p. 302)</a></li> </ul>	November 06, 2017
EU (London) Region	Amazon S3 is now available in the EU (London) Region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	December 13, 2016
Canada (Central) Region	Amazon S3 is now available in the Canada (Central) Region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	December 8, 2016
Object tagging support	<p>Amazon S3 now supports object tagging. The following new API operations support object tagging:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Object tagging (p. 454)</a></li> <li>• <a href="#">GET Object tagging (p. 368)</a></li> <li>• <a href="#">DELETE Object tagging (p. 347)</a></li> </ul> <p>In addition, other API operations are updated to support object tagging. For more information, see <a href="#">Object Tagging</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	November 29, 2016
S3 lifecycle now supports object tag based filter	<p>Amazon S3 now supports tag-based filtering in lifecycle configuration. You can now specify a lifecycle rule, in which you can specify a key prefix, one or more object tags, or a combination of both, to select a subset of objects to which the lifecycle rule applies. For more information, see <a href="#">Object Lifecycle Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Amazon S3 now supports Expedited and Bulk data retrievals in addition to Standard retrievals when restoring objects archived to Glacier.</p>	November 29, 2016

Change	Description	Release Date
CloudWatch request metrics for buckets	<p>Amazon S3 now supports CloudWatch metrics for requests made on buckets. The following new API operations support configuring request metrics:</p> <ul style="list-style-type: none"> <li>• <a href="#">DELETE Bucket metrics (p. 90)</a></li> <li>• <a href="#">GET Bucket metrics (p. 160)</a></li> <li>• <a href="#">PUT Bucket metrics (p. 285)</a></li> <li>• <a href="#">List Bucket Metrics Configurations (p. 215)</a></li> </ul> <p>For more information, see <a href="#">Monitoring Metrics with Amazon CloudWatch</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	November 29, 2016
Amazon S3 Inventory	<p>Amazon S3 now supports storage inventory. Amazon S3 inventory provides a flat-file output of your objects and their corresponding metadata on a daily or weekly basis for an S3 bucket or a shared prefix (that is, objects that have names that begin with a common string).</p> <p>The following new API operations are for storage inventory:</p> <ul style="list-style-type: none"> <li>• <a href="#">DELETE Bucket inventory (p. 86)</a></li> <li>• <a href="#">GET Bucket Inventory (p. 139)</a></li> <li>• <a href="#">PUT Bucket inventory (p. 258)</a></li> <li>• <a href="#">List Bucket Inventory Configurations (p. 210)</a></li> </ul> <p>For more information, see <a href="#">Amazon S3 Storage Inventory</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	November 29, 2016

Change	Description	Release Date
Amazon S3 Analytics – Storage Class Analysis	<p>The new Amazon S3 analytics – storage class analysis feature observes data access patterns to help you determine when to transition less frequently accessed STANDARD storage to the STANDARD_IA (IA, for infrequent access) storage class. After storage class analysis observes the infrequent access patterns of a filtered set of data over a period of time, you can use the analysis results to help you improve your lifecycle policies. This feature also includes a detailed daily analysis of your storage usage at the specified bucket, prefix, or tag level that you can export to a S3 bucket.</p> <p>The following new API operations are for storage class analysis:</p> <ul style="list-style-type: none"> <li>• <a href="#">DELETE Bucket analytics (p. 80)</a></li> <li>• <a href="#">GET Bucket analytics (p. 126)</a></li> <li>• <a href="#">PUT Bucket analytics (p. 242)</a></li> <li>• <a href="#">List Bucket Analytics Configurations (p. 206)</a></li> </ul> <p>For more information, see <a href="#">Amazon S3 Analytics – Storage Class Analysis</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	November 29, 2016
<a href="#">Added Glacier retrieval options to POST Object restore (p. 397)</a>	Amazon S3 now supports Expedited and Bulk data retrievals in addition to Standard retrievals when restoring objects archived to Glacier. For more information, see <a href="#">Restoring Archived Objects</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	November 21, 2016
US East (Ohio) Region	Amazon S3 is now available in the US East (Ohio) Region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	October 17, 2016
Asia Pacific (Mumbai) region	Amazon S3 is now available in the Asia Pacific (Mumbai) region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	June 27, 2016
GET Bucket (List Objects) API revised	The GET Bucket (List Objects) API has been revised. We recommend that you use the new version, GET Bucket (List Objects) version 2. For more information, see <a href="#">GET Bucket (List Objects) Version 2 (p. 101)</a> .	May 4, 2016

Change	Description	Release Date
Amazon S3 Transfer Acceleration	<p>Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations.</p> <p>For more information, see <a href="#">Transfer Acceleration</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following new API operations support Transfer Acceleration: <a href="#">GET Bucket accelerate (p. 120)</a> and <a href="#">PUT Bucket accelerate (p. 232)</a>.</p>	April 19, 2016
Lifecycle support to remove expired object delete marker	<p>Lifecycle configuration expiration action now allows you to direct Amazon S3 to remove expired object delete markers in versioned bucket. For more information, see <a href="#">Elements to Describe Lifecycle Actions</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	March 16, 2016
Bucket lifecycle configuration now supports the action to abort incomplete multipart uploads	<p>Bucket lifecycle configuration now supports the <code>AbortIncompleteMultipartUpload</code> action that you can use to direct Amazon S3 to abort multipart uploads that don't complete within a specified number of days after being initiated. When a multipart upload becomes eligible for an abort operation, Amazon S3 deletes any uploaded parts and aborts the multipart upload.</p> <p>The following API operations have been updated to support the new action:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Bucket lifecycle (p. 265)</a> – The XML configuration now allows you to specify the <code>AbortIncompleteMultipartUpload</code> action in a lifecycle configuration rule.</li> <li>• <a href="#">List Parts (p. 502)</a> and <a href="#">Initiate Multipart Upload (p. 492)</a> – Both of these API operations now return two additional response headers (<code>x-amz-abort-date</code>, and <code>x-amz-abort-rule-id</code>) if the bucket has a lifecycle rule that specifies the <code>AbortIncompleteMultipartUpload</code> action. These headers in the response indicate when the initiated multipart upload will become eligible for an abort operation and which lifecycle rule is applicable.</li> </ul> <p>For conceptual information, see the following topics in the <i>Amazon Simple Storage Service Developer Guide</i>:</p> <ul style="list-style-type: none"> <li>• <a href="#">Aborting Incomplete Multipart Uploads Using a Bucket Lifecycle Policy</a></li> <li>• <a href="#">Elements to Describe Lifecycle Actions</a></li> </ul>	March 16, 2016

Change	Description	Release Date
Amazon S3 Signature Version 4 now supports unsigned payloads	Amazon S3 Signature Version 4 now supports unsigned payloads when authenticating requests using the Authorization header. Because you don't sign the payload, it does not provide the same security that comes with payload signing, but it provides similar performance characteristics as signature version 2. For more information, see <a href="#">Signature Calculations for the Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4)</a> (p. 18).	January 15, 2016
Asia Pacific (Seoul) region	Amazon S3 is now available in the Asia Pacific (Seoul) region. For more information about Amazon S3 regions and endpoints, see <a href="#">Regions and Endpoints</a> in the <i>AWS General Reference</i> .	January 6, 2016
Renamed the US Standard region	Changed the region name string from US Standard to US East (N. Virginia). This is only a region name update, there is no change in the functionality.	December 11, 2015
New storage class	<p>Amazon S3 now offers a new storage class, STANDARD_IA (IA, for infrequent access) for storing objects. This storage class is optimized for long-lived and less frequently accessed data. For more information, see <a href="#">Storage Classes</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Lifecycle configuration feature updates now allow you to transition objects to the STANDARD_IA storage class. For more information, see <a href="#">Object Lifecycle Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Previously, the cross-region replication feature used the storage class of the source object for object replicas. Now, when you configure cross-region replication you can specify a storage class for the object replica created in the destination bucket. For more information, see <a href="#">Cross-Region Replication</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	September 16, 2015
Event notifications	Amazon S3 event notifications have been updated to add notifications when objects are deleted and to add filtering on object names with prefix and suffix matching. For the relevant API operations, see <a href="#">PUT Bucket notification</a> (p. 290), and <a href="#">GET Bucket notification</a> (p. 164). For more information, see <a href="#">Configuring Amazon S3 Event Notifications</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	July 28, 2015
Cross-region replication	Amazon S3 now supports cross-region replication. Cross-region replication is the automatic, asynchronous copying of objects across buckets in different AWS regions. For the relevant API operations, see <a href="#">PUT Bucket replication</a> (p. 302), <a href="#">GET Bucket replication</a> (p. 187) and <a href="#">DELETE Bucket replication</a> (p. 95). For more information, see <a href="#">Enabling Cross-Region Replication</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	March 24, 2015

Change	Description	Release Date
Event notifications	<p>Amazon S3 now supports new event types and destinations in a bucket notification configuration. Prior to this release, Amazon S3 supported only the <code>s3:ReducedRedundancyLostObject</code> event type and an Amazon SNS topic as the destination. For more information about the new event types, go to <a href="#">Setting Up Notification of Bucket Events</a> in the <i>Amazon Simple Storage Service Developer Guide</i>. For the relevant API operations, see <a href="#">PUT Bucket notification</a> (p. 290) and <a href="#">GET Bucket notification</a> (p. 164).</p>	November 13, 2014
Server-side encryption with AWS Key Management Service (KMS)	<p>Amazon S3 now supports server-side encryption using AWS Key Management Service (KMS). With server-side encryption with KMS, you manage the envelope key through KMS, and Amazon S3 calls KMS to access the envelope key within the permissions you set.</p> <p>For more information about server-side encryption with KMS, see <a href="#">Protecting Data Using Server-Side Encryption with AWS Key Management Service</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following Amazon S3 REST API operations support headers related to KMS.</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Object</a> (p. 412)</li> <li>• <a href="#">PUT Object - Copy</a> (p. 431)</li> <li>• <a href="#">POST Object</a> (p. 385)</li> <li>• <a href="#">Initiate Multipart Upload</a> (p. 492)</li> <li>• <a href="#">Upload Part</a> (p. 508)</li> </ul>	November 12, 2014
EU (Frankfurt) region	Amazon S3 is now available in the EU (Frankfurt) region.	October 23, 2014

Change	Description	Release Date
Server-side encryption with customer-provided encryption keys	<p>Amazon S3 now supports server-side encryption using customer-provided encryption keys (SSE-C). Server-side encryption enables you to request Amazon S3 to encrypt your data at rest. When using SSE-C, Amazon S3 encrypts your objects with the custom encryption keys that you provide. Since Amazon S3 performs the encryption for you, you get the benefits of using your own encryption keys without the cost of writing or executing your own encryption code.</p> <p>For more information about SSE-C, go to <a href="#">Server-Side Encryption (Using Customer-Provided Encryption Keys)</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The following Amazon S3 REST API operations support headers related to SSE-C.</p> <ul style="list-style-type: none"> <li>• <a href="#">GET Object (p. 349)</a></li> <li>• <a href="#">HEAD Object (p. 373)</a></li> <li>• <a href="#">PUT Object (p. 412)</a></li> <li>• <a href="#">PUT Object - Copy (p. 431)</a></li> <li>• <a href="#">POST Object (p. 385)</a></li> <li>• <a href="#">Initiate Multipart Upload (p. 492)</a></li> <li>• <a href="#">Upload Part (p. 508)</a></li> <li>• <a href="#">Upload Part - Copy (p. 514)</a></li> </ul>	June 12, 2014
Lifecycle support for versioning	<p>Prior to this release lifecycle configuration was supported only on nonversioned buckets. Now you can configure lifecycle on both the nonversioned and versioning-enabled buckets.</p> <p>For more information, go to <a href="#">Object Lifecycle Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The related API operations, see <a href="#">PUT Bucket lifecycle (p. 265)</a>, <a href="#">GET Bucket lifecycle (p. 145)</a>, and <a href="#">DELETE Bucket lifecycle (p. 88)</a>.</p>	May 20, 2014
Amazon S3 now supports Signature Version 4	<p>Amazon S3 now supports Signature Version 4 (SigV4) in all regions, the latest specification for how to sign and authenticate AWS requests.</p> <p>For more information, see <a href="#">Authenticating Requests (AWS Signature Version 4) (p. 14)</a>.</p>	January 30, 2014

Change	Description	Release Date
Amazon S3 list actions now support encoding-type request parameter	<p>The following Amazon S3 list actions now support encoding-type optional request parameter.</p> <p><a href="#">GET Bucket (List Objects) Version 1 (p. 111)</a></p> <p><a href="#">GET Bucket Object versions (p. 173)</a></p> <p><a href="#">List Multipart Uploads (p. 218)</a></p> <p><a href="#">List Parts (p. 502)</a></p> <p>An object key can contain any Unicode character; however, the XML 1.0 parser cannot parse some characters, such as characters with an ASCII value from 0 to 10. For characters that are not supported in XML 1.0, you can add this parameter to request that Amazon S3 encode the keys in the response.</p>	November 1, 2013
SOAP Support Over HTTP Deprecated	SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.	September 19, 2013
Root domain support for website hosting	<p>Amazon S3 now supports hosting static websites at the root domain. Visitors to your website can access your site from their browser without specifying "www" in the web address (e.g., "example.com"). Many customers already host static websites on Amazon S3 that are accessible from a "www" subdomain (e.g., "www.example.com"). Previously, to support root domain access, you needed to run your own web server to proxy root domain requests from browsers to your website on Amazon S3. Running a web server to proxy requests introduces additional costs, operational burden, and another potential point of failure. Now, you can take advantage of the high availability and durability of Amazon S3 for both "www" and root domain addresses.</p> <p>For an example walkthrough, go to <a href="#">Example: Setting Up a Static Website Using a Custom Domain</a> in the <i>Amazon Simple Storage Service Developer Guide</i>. For conceptual information, go to <a href="#">Hosting Static Websites on Amazon S3</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	December 27, 2012

Change	Description	Release Date
Support for Archiving Data to Amazon Glacier	<p>Amazon S3 now supports a storage option that enables you to utilize Amazon Glacier's low-cost storage service for data archival. To archive objects, you define archival rules identifying objects and a timeline when you want Amazon S3 to archive these objects to Glacier. You can easily set the rules on a bucket using the Amazon S3 console or programmatically using the Amazon S3 API or AWS SDKs.</p> <p>To support data archival rules, Amazon S3 lifecycle management API has been updated. For more information, see <a href="#">PUT Bucket lifecycle (p. 265)</a>.</p> <p>After you archive objects, you must first restore a copy before you can access the data. Amazon S3 offers a new API for you to initiate a restore. For more information, see <a href="#">POST Object restore (p. 397)</a>.</p> <p>For conceptual information, go to <a href="#">Object Lifecycle Management</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	November 13, 2012
Support for Website Page Redirects	<p>For a bucket that is configured as a website, Amazon S3 now supports redirecting a request for an object to another object in the same bucket or to an external URL. You can configure redirect by adding the <code>x-amz-website-redirect-location</code> metadata to the object.</p> <p>The object upload API operations <a href="#">PUT Object (p. 412)</a>, <a href="#">Initiate Multipart Upload (p. 492)</a>, and <a href="#">POST Object (p. 385)</a> allow you to configure the <code>x-amz-website-redirect-location</code> object metadata.</p> <p>For conceptual information, go to <a href="#">How to Configure Website Page Redirects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	October 4, 2012
Cross-Origin Resource Sharing (CORS) support	<p>Amazon S3 now supports Cross-Origin Resource Sharing (CORS). CORS defines a way in which client web applications that are loaded in one domain can interact with or access resources in a different domain. With CORS support in Amazon S3, you can build rich client-side web applications on top of Amazon S3 and selectively allow cross-domain access to your Amazon S3 resources. For more information, see <a href="#">Enabling Cross-Origin Resource Sharing</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	August 31, 2012
Cost Allocation Tagging support	<p>Amazon S3 now supports cost allocation tagging, which allows you to label S3 buckets so you can more easily track their cost against projects or other criteria. For more information, see <a href="#">Cost Allocation Tagging</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	August 21, 2012

Change	Description	Release Date
Object Expiration support	You can use Object Expiration to schedule automatic removal of data after a configured time period. You set object expiration by adding lifecycle configuration to a bucket. For more information, see <a href="#">Transitioning Objects: General Considerations</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	December 27, 2011
New Region supported	Amazon S3 now supports the South America (São Paulo) region. For more information, see <a href="#">Buckets and Regions</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	December 14, 2011
Multi-Object Delete	<p>Amazon S3 now supports Multi-Object Delete API that enables you to delete multiple objects in a single request. With this feature, you can remove large numbers of objects from Amazon S3 more quickly than using multiple individual DELETE requests.</p> <p>For more information about the API see, see <a href="#">Delete Multiple Objects (p. 333)</a>.</p> <p>For conceptual information about the delete operation, see <a href="#">Deleting Objects</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	December 7, 2011
New region supported	Amazon S3 now supports the US West (Oregon) region. For more information, see <a href="#">Buckets and Regions</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	November 8, 2011
Server-side encryption support	Amazon S3 now supports server-side encryption. It enables you to request Amazon S3 to encrypt your data at rest, that is, encrypt your object data when Amazon S3 writes your data to disks in its data centers. To request server-side encryption, you must add the <code>x-amz-server-side-encryption</code> header to your request. To learn more about data encryption, go to <a href="#">Using Data Encryption</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	October 17, 2011
Multipart Upload API extended to enable copying objects up to 5 TB	Prior to this release, Amazon S3 API supported copying objects (see <a href="#">PUT Object - Copy (p. 431)</a> ) of up to 5 GB in size. To enable copying objects larger than 5 GB, Amazon S3 extends the multipart upload API with a new operation, <a href="#">Upload Part (Copy)</a> . You can use this multipart upload operation to copy objects up to 5 TB in size. For conceptual information about multipart upload, go to <a href="#">Uploading Objects Using Multipart Upload</a> in the <i>Amazon Simple Storage Service Developer Guide</i> . To learn more about the new API, see <a href="#">Upload Part - Copy (p. 514)</a> .	June 21, 2011
SOAP API calls over HTTP disabled	To increase security, SOAP API calls over HTTP are disabled. Authenticated and anonymous SOAP requests must be sent to Amazon S3 using SSL.	June 6, 2011

Change	Description	Release Date
Support for hosting static websites in Amazon S3	<p>Amazon S3 introduces enhanced support for hosting static websites. This includes support for index documents and custom error documents. When using these features, requests to the root of your bucket or a subfolder (e.g., <code>http://mywebsite.com/subfolder</code>) returns your index document instead of the list of objects in your bucket. If an error is encountered, Amazon S3 returns your custom error message instead of an Amazon S3 error message. For API information to configure your bucket as a website, see the following sections:</p> <ul style="list-style-type: none"> <li>• <a href="#">PUT Bucket website (p. 321)</a></li> <li>• <a href="#">GET Bucket website (p. 202)</a></li> <li>• <a href="#">DELETE Bucket website (p. 99)</a></li> </ul> <p>For conceptual overview, go to <a href="#">Hosting Websites on Amazon S3</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p>	February 17, 2011
Response Header API Support	The GET Object REST API now allows you to change the response headers of the REST GET Object request for each request. That is, you can alter object metadata in the response, without altering the object itself. For more information, see <a href="#">GET Object (p. 349)</a> .	January 14, 2011
Large Object Support	Amazon S3 has increased the maximum size of an object you can store in an S3 bucket from 5 GB to 5 TB. If you are using the REST API you can upload objects of up to 5 GB size in a single PUT operation. For larger objects, you must use the Multipart Upload REST API to upload objects in parts. For conceptual information, go to <a href="#">Uploading Objects Using Multipart Upload</a> in the <i>Amazon Simple Storage Service Developer Guide</i> . For multipart upload API information, see <a href="#">Initiate Multipart Upload (p. 492)</a> , <a href="#">Upload Part (p. 508)</a> , <a href="#">Complete Multipart Upload (p. 486)</a> , <a href="#">List Parts (p. 502)</a> , and <a href="#">List Multipart Uploads (p. 218)</a>	December 9, 2010
Multipart upload	Multipart upload enables faster, more flexible uploads into Amazon S3. It allows you to upload a single object as a set of parts. For conceptual information, go to <a href="#">Uploading Objects Using Multipart Upload</a> in the <i>Amazon Simple Storage Service Developer Guide</i> . For multipart upload API information, see <a href="#">Initiate Multipart Upload (p. 492)</a> , <a href="#">Upload Part (p. 508)</a> , <a href="#">Complete Multipart Upload (p. 486)</a> , <a href="#">List Parts (p. 502)</a> , and <a href="#">List Multipart Uploads (p. 218)</a>	November 10, 2010
Notifications	The Amazon S3 notifications feature enables you to configure a bucket so that Amazon S3 publishes a message to an Amazon Simple Notification Service (SNS) topic when Amazon S3 detects a key event on a bucket. For more information, see <a href="#">GET Bucket notification (p. 164)</a> and <a href="#">PUT Bucket notification (p. 164)</a> .	July 14, 2010

Change	Description	Release Date
Bucket policies	Bucket policies is an access management system you use to set access permissions on buckets, objects, and sets of objects. This functionality supplements and in many cases replaces access control lists.	July 6, 2010
Reduced Redundancy	Amazon S3 now enables you to reduce your storage costs by storing objects in Amazon S3 with reduced redundancy. For more information, see <a href="#">PUT Object (p. 412)</a> .	May 12, 2010
New region supported	Amazon S3 now supports the Asia Pacific (Singapore) region and therefore new location constraints. For more information, see <a href="#">GET Bucket location (p. 152)</a> and <a href="#">PUT Bucket (p. 227)</a> .	April 28, 2010
Object Versioning	This release introduces object Versioning. All objects now have a key and a version. If you enable versioning for a bucket, Amazon S3 gives all objects added to a bucket a unique version ID. This feature enables you to recover from unintended overwrites and deletions. For more information, see <a href="#">GET Object (p. 349)</a> , <a href="#">DELETE Object (p. 343)</a> , <a href="#">PUT Object (p. 412)</a> , <a href="#">PUT Object Copy (p. 431)</a> , or <a href="#">POST Object (p. 385)</a> . The SOAP API does not support versioned objects.	February 8, 2010
New region supported	Amazon S3 now supports the US-West (Northern California) region. The new endpoint is <code>s3-us-west-1.amazonaws.com</code> . For more information, see <a href="#">How to Select a Region for Your Buckets</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .	December 2, 2009
C# Library Support	AWS now provides Amazon S3 C# libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific API operations instead of REST or SOAP. These libraries provide basic functions (not included in the REST or SOAP APIs), such as request authentication, request retries, and error handling so that it's easier to get started.	November 11, 2009
Technical documents reorganized	The API reference has been split out of the <i>Amazon S3 Developer Guide</i> . Now, on the documentation landing page, <a href="#">Amazon Simple Storage Service Documentation</a> , you can select the document you want to view. When viewing the documents online, the links in one document will take you, when appropriate, to one of the other guides.	September 16, 2009

# Appendix

## Topics

- [Appendix: SOAP API \(p. 541\)](#)
- [Appendix: Lifecycle Configuration APIs \(Deprecated\) \(p. 568\)](#)

## Appendix: SOAP API

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes the SOAP API with respect to service, bucket, and object operations. Note that SOAP requests, both authenticated and anonymous, must be sent to Amazon S3 using SSL. Amazon S3 returns an error when you send a SOAP request over HTTP.

The latest Amazon S3 WSDL is available at <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>.

## Topics

- [Operations on the Service \(SOAP API\) \(p. 541\)](#)
- [Operations on Buckets \(SOAP API\) \(p. 542\)](#)
- [Operations on Objects \(SOAP API\) \(p. 551\)](#)
- [SOAP Error Responses \(p. 566\)](#)

## Operations on the Service (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on the Amazon S3 service.

## Topics

- [ListAllMyBuckets \(SOAP API\) \(p. 541\)](#)

## ListAllMyBuckets (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `ListAllMyBuckets` operation returns a list of all buckets owned by the sender of the request.

## Example

### Sample Request

```
<ListAllMyBuckets xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListAllMyBuckets>
```

#### Sample Response

```
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
<Owner>
<ID>bcaf1ffd86f41161ca5fb16fd081034f</ID>
<DisplayName>webfile</DisplayName>
</Owner>
<Buckets>
<Bucket>
<Name>quotes;</Name>
<CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
</Bucket>
<Bucket>
<Name>samples</Name>
<CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

### Response Body

- **Owner:**

This provides information that Amazon S3 uses to represent your identity for purposes of authentication and access control. ID is a unique and permanent identifier for the developer who made the request. DisplayName is a human-readable name representing the developer who made the request. It is not unique, and might change over time. We recommend that you match your DisplayName to your Forum name.

- **Name:**

The name of a bucket. Note that if one of your buckets was recently deleted, the name of the deleted bucket might still be present in this list for a period of time.

- **CreationDate:**

The time that the bucket was created.

### Access Control

You must authenticate with a valid AWS Access Key ID. Anonymous requests are never allowed to list buckets, and you can only list buckets for which you are the owner.

## Operations on Buckets (SOAP API)

#### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on Amazon S3 buckets.

#### Topics

- [CreateBucket \(SOAP API\) \(p. 543\)](#)
- [DeleteBucket \(SOAP API\) \(p. 544\)](#)
- [ListBucket \(SOAP API\) \(p. 544\)](#)
- [GetBucketAccessControlPolicy \(SOAP API\) \(p. 547\)](#)
- [SetBucketAccessControlPolicy \(SOAP API\) \(p. 548\)](#)
- [GetBucketLoggingStatus \(SOAP API\) \(p. 549\)](#)
- [SetBucketLoggingStatus \(SOAP API\) \(p. 550\)](#)

## CreateBucket (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `CreateBucket` operation creates a bucket. Not every string is an acceptable bucket name. For information on bucket naming restrictions, see [Working with Amazon S3 Buckets](#).

### Note

To determine whether a bucket name exists, use `ListBucket` and set `MaxKeys` to 0. A `NoSuchBucket` response indicates that the bucket is available, an `AccessDenied` response indicates that someone else owns the bucket, and a `Success` response indicates that you own the bucket or have permission to access it.

### Example Create a bucket named "quotes"

#### Sample Request

```
<CreateBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</CreateBucket>
```

#### Sample Response

```
<CreateBucketResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <CreateBucketResponse>
    <Bucket>quotes</Bucket>
  </CreateBucketResponse>
</CreateBucketResponse>
```

## Elements

- **Bucket**: The name of the bucket you are trying to create.
- **AccessControlList**: The access control list for the new bucket. This element is optional. If not provided, the bucket is created with an access policy that give the requester `FULL_CONTROL` access.

## Access Control

You must authenticate with a valid AWS Access Key ID. Anonymous requests are never allowed to create buckets.

## Related Resources

- [ListBucket \(SOAP API\) \(p. 544\)](#)

## DeleteBucket (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The DeleteBucket operation deletes a bucket. All objects in the bucket must be deleted before the bucket itself can be deleted.

### Example

This example deletes the "quotes" bucket.

#### Sample Request

```
<DeleteBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <AWSAccessKeyId> AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</DeleteBucket>
```

#### Sample Response

```
<DeleteBucketResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <DeleteBucketResponse>
    <Code>204</Code>
    <Description>No Content</Description>
  </DeleteBucketResponse>
</DeleteBucketResponse>
```

## Elements

- **Bucket**: The name of the bucket you want to delete.

## Access Control

Only the owner of a bucket is allowed to delete it, regardless the access control policy on the bucket.

## ListBucket (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The ListBucket operation returns information about some of the items in the bucket.

For a general introduction to the list operation, see the [Listing Object Keys](#).

## Requests

This example lists up to 1000 keys in the "quotes" bucket that have the prefix "notes."

## Syntax

```
<ListBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Prefix>notes/</Prefix>
  <Delimiter>/<Delimiter>
  <MaxKeys>1000</MaxKeys>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

## Parameters

Name	Description	Required
<code>prefix</code>	Limits the response to keys which begin with the indicated prefix. You can use prefixes to separate a bucket into different sets of keys in a way similar to how a file system uses folders.  Type: String  Default: None	No
<code>marker</code>	Indicates where in the bucket to begin listing. The list will only include keys that occur lexicographically after marker. This is convenient for pagination: To get the next page of results use the last key of the current page as the marker.  Type: String  Default: None	No
<code>max-keys</code>	The maximum number of keys you'd like to see in the response body. The server might return fewer than this many keys, but will not return more.  Type: String  Default: None	No
<code>delimiter</code>	Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.  Type: String  Default: None	No

## Success Response

This response assumes the bucket contains the following keys:

```
notes/todos.txt
notes/2005-05-23/customer_mtg_notes.txt
```

```
notes/2005-05-23/phone_notes.txt
notes/2005-05-28/sales_notes.txt
```

## Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>backups</Name>
  <Prefix>notes/</Prefix>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>notes/todos.txt</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fd96f00ad9f27c383fc9ac7f"</ETag>
    <Size>5126</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>75aa57f09aa0c8caeab4f8c24e99d10f8e7faeefbf76c078efc7c6caea54ba06a</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>notes/2005-05-23/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>notes/2005-05-28/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

As you can see, many of the fields in the response echo the request parameters. `IsTruncated`, `Contents`, and `CommonPrefixes` are the only response elements that can contain new information.

## Response Elements

Name	Description
Contents	<p>Metadata about each object returned.</p> <p>Type: XML metadata</p> <p>Ancestor: <code>ListBucketResult</code></p>
CommonPrefixes	<p>A response can contain <code>CommonPrefixes</code> only if you specify a <code>delimiter</code>. When you do, <code>CommonPrefixes</code> contains all (if there are any) keys between <code>Prefix</code> and the next occurrence of the string specified by <code>delimiter</code>. In effect, <code>CommonPrefixes</code> lists keys that act like subdirectories in the directory specified by <code>Prefix</code>. For example, if <code>prefix</code> is <code>notes/</code> and <code>delimiter</code> is a slash (/), in <code>notes/summer/july</code>, the common prefix is <code>notes/summer/</code>.</p> <p>Type: String</p> <p>Ancestor: <code>ListBucketResult</code></p>
Delimiter	<p>Causes keys that contain the same string between the prefix and the first occurrence of the delimiter to be rolled up into a single result element in the <code>CommonPrefixes</code> collection. These rolled-up keys are not returned elsewhere in the response.</p>

Name	Description
	Type: String  Ancestor: ListBucketResult
IsTruncated	Specifies whether (true) or not (false) all of the results were returned. All of the results may not be returned if the number of results exceeds that specified by MaxKeys.  Type: String  Ancestor: boolean
Marker	Indicates where in the bucket to begin listing.  Type: String  Ancestor: ListBucketResult
MaxKeys	The maximum number of keys returned in the response body.  Type: String  Ancestor: ListBucketResult
Name	Name of the bucket.  Type: String  Ancestor: ListBucketResult
Prefix	Keys that begin with the indicated prefix.  Type: String  Ancestor: ListBucketResult

## Response Body

For information about the list response, see [Listing Keys Response](#).

## Access Control

To list the keys of a bucket you need to have been granted `READ` access on the bucket.

## GetBucketAccessControlPolicy (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `GetBucketAccessControlPolicy` operation fetches the access control policy for a bucket.

### Example

This example retrieves the access control policy for the "quotes" bucket.

#### Sample Request

```
<GetBucketAccessControlPolicy xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<Bucket>quotes</Bucket>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetBucketAccessControlPolicy>
```

#### Sample Response

```
<AccessControlPolicy>
<Owner>
  <ID>a9a7b886d6fd2441bf9b1c61be666e9</ID>
  <DisplayName>chriscustomer</DisplayName>
</Owner>
<AccessControlList>
  <Grant>
    <Grantee xsi:type="CanonicalUser">
      <ID>a9a7b886d6f41bf9b1c61be666e9</ID>
      <DisplayName>chriscustomer</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
  <Grant>
    <Grantee xsi:type="Group">
      <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
    </Grantee>
    <Permission>READ</Permission>
  </Grant>
</AccessControlList>
<AccessControlPolicy>
```

## Response Body

The response contains the access control policy for the bucket. For an explanation of this response, see [SOAP Access Policy](#).

## Access Control

You must have `READ_ACP` rights to the bucket in order to retrieve the access control policy for a bucket.

## SetBucketAccessControlPolicy (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `SetBucketAccessControlPolicy` operation sets the Access Control Policy for an existing bucket. If successful, the previous Access Control Policy for the bucket is entirely replaced with the specified Access Control Policy.

### Example

Give the specified user (usually the owner) `FULL_CONTROL` access to the "quotes" bucket.

#### Sample Request

```
<SetBucketAccessControlPolicy xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<Bucket>quotes</Bucket>
```

```
<AccessControlList>
  <Grant>
    <Grantee xsi:type="CanonicalUser">
      <ID>a9a7b8863000e241bf9b1c61be666e9</ID>
      <DisplayName>chriscustomer</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</SetBucketAccessControlPolicy >
```

#### Sample Response

```
<GetBucketAccessControlPolicyResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <GetBucketAccessControlPolicyResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </GetBucketAccessControlPolicyResponse>
</GetBucketAccessControlPolicyResponse>
```

## Access Control

You must have `WRITE_ACP` rights to the bucket in order to set the access control policy for a bucket.

## GetBucketLoggingStatus (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `GetBucketLoggingStatus` retrieves the logging status for an existing bucket.

For a general introduction to this feature, see [Server Logs](#).

### Example

#### Sample Request

```
<?xml version="1.0" encoding="utf-8"?>
  <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <ns1:Body xmlns:ns1="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <GetBucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
        <Bucket>mybucket</Bucket>
        <AWSAccessKeyId>YOUR_AWS_ACCESS_KEY_ID</AWSAccessKeyId>
        <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
        <Signature>YOUR_SIGNATURE_HERE</Signature>
      </GetBucketLoggingStatus>
    </ns1:Body>
  </soap:Envelope>
```

#### Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
    instance">  
    <soapenv:Header>  
    </soapenv:Header>  
    <soapenv:Body>  
        <GetBucketLoggingStatusResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">  
            <GetBucketLoggingStatusResponse>  
                <LoggingEnabled>  
                    <TargetBucket>mylogs</TargetBucket>  
                    <TargetPrefix>mybucket-access_log-</TargetPrefix>  
                </LoggingEnabled>  
            </GetBucketLoggingStatusResponse>  
        </GetBucketLoggingStatusResponse>  
    </soapenv:Body>  
</soapenv:Envelope>
```

## Access Control

Only the owner of a bucket is permitted to invoke this operation.

## SetBucketLoggingStatus (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The SetBucketLoggingStatus operation updates the logging status for an existing bucket.

For a general introduction to this feature, see [Server Logs](#).

### Example

This sample request enables server access logging for the 'mybucket' bucket, and configures the logs to be delivered to 'mylogs' under prefix 'access\_log-'.

#### Sample Request

```
<?xml version="1.0" encoding="utf-8"?>  
    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"  
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/  
        XMLSchema">  
        <soap:Body>  
            <SetBucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">  
                <Bucket>myBucket</Bucket>  
                <AWSAccessKeyId>YOUR_AWS_ACCESS_KEY_ID</AWSAccessKeyId>  
                <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>  
                <Signature>YOUR_SIGNATURE_HERE</Signature>  
                <BucketLoggingStatus>  
                    <LoggingEnabled>  
                        <TargetBucket>mylogs</TargetBucket>  
                        <TargetPrefix>mybucket-access_log-</TargetPrefix>  
                    </LoggingEnabled>  
                </BucketLoggingStatus>  
            </SetBucketLoggingStatus>  
        </soap:Body>  
    </soap:Envelope>
```

#### Sample Response

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" >
    <soapenv:Header>
    </soapenv:Header>
    <soapenv:Body>
        <SetBucketLoggingStatusResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01"/>
    </soapenv:Body>
</soapenv:Envelope>
```

## Access Control

Only the owner of a bucket is permitted to invoke this operation.

# Operations on Objects (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

This section describes operations you can perform on Amazon S3 objects.

### Topics

- [PutObjectInline \(SOAP API\) \(p. 551\)](#)
- [PutObject \(SOAP API\) \(p. 553\)](#)
- [CopyObject \(SOAP API\) \(p. 555\)](#)
- [GetObject \(SOAP API\) \(p. 559\)](#)
- [GetObjectExtended \(SOAP API\) \(p. 563\)](#)
- [DeleteObject \(SOAP API\) \(p. 564\)](#)
- [GetObjectAccessControlPolicy \(SOAP API\) \(p. 565\)](#)
- [SetObjectAccessControlPolicy \(SOAP API\) \(p. 566\)](#)

## PutObjectInline (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `PutObjectInline` operation adds an object to a bucket. The data for the object is provided in the body of the SOAP message.

If an object already exists in a bucket, the new object will overwrite it because Amazon S3 stores the last write request. However, Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object nearly simultaneously, all of the objects might be stored, even though only one wins in the end. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer.

To ensure an object is not corrupted over the network, you can calculate the MD5 of an object, PUT it to Amazon S3, and compare the returned Etag to the calculated MD5 value.

PutObjectInline is not suitable for use with large objects. The system limits this operation to working with objects 1MB or smaller. PutObjectInline will fail with the `InlineDataTooLargeError` status code if the Data parameter encodes an object larger than 1MB. To upload large objects, consider using the non-inline PutObject API, or the REST API instead.

### Example

This example writes some text and metadata into the "Nelson" object in the "quotes" bucket, give a user (usually the owner) `FULL_CONTROL` access to the object, and make the object readable by anonymous parties.

#### Sample Request

```
<PutObjectInline xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Metadata>
    <Name>family</Name>
    <Value>Muntz</Value>
  </Metadata>
  <Data>aGEtaGE=</Data>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b886d6fde241bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObjectInline>
```

#### Sample Response

```
<PutObjectInlineResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <PutObjectInlineResponse>
    <ETag>"828ef3fdfa96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2006-01-01T12:00:00.000Z</lastModified>
  </PutObjectInlineResponse>
</PutObjectInlineResponse>
```

## Elements

- **Bucket:** The bucket in which to add the object.
- **Key:** The key to assign to the object.
- **Metadata:** You can provide name-value metadata pairs in the metadata element. These will be stored with the object.

- **Data**: The base 64 encoded form of the data.
- **ContentLength**: The length of the data in bytes.
- **AccessControlList**: An Access Control List for the resource. This element is optional. If omitted, the requester is given FULL\_CONTROL access to the object. If the object already exists, the preexisting access control policy is replaced.

## Responses

- **ETag**: The entity tag is an MD5 hash of the object that you can use to do conditional fetches of the object using `GetObjectExtended`. The ETag only reflects changes to the contents of an object, not its metadata.
- **LastModified**: The Amazon S3 timestamp for the saved object.

## Access Control

You must have `WRITE` access to the bucket in order to put objects into the bucket.

## Related Resources

- [PutObject \(SOAP API\) \(p. 553\)](#)
- [CopyObject \(SOAP API\) \(p. 555\)](#)

## PutObject (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `PutObject` operation adds an object to a bucket. The data for the object is attached as a DIME attachment.

To ensure an object is not corrupted over the network, you can calculate the MD5 of an object, PUT it to Amazon S3, and compare the returned Etag to the calculated MD5 value.

If an object already exists in a bucket, the new object will overwrite it because Amazon S3 stores the last write request. However, Amazon S3 is a distributed system. If Amazon S3 receives multiple write requests for the same object nearly simultaneously, all of the objects might be stored, even though only one wins in the end. Amazon S3 does not provide object locking; if you need this, make sure to build it into your application layer.

### Example

This example puts some data and metadata in the "Nelson" object of the "quotes" bucket, give a user (usually the owner) `FULL_CONTROL` access to the object, and make the object readable by anonymous parties. In this sample, the actual attachment is not shown.

#### Sample Request

```
<PutObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
```

```
<Value>text/plain</Value>
</Metadata>
<Metadata>
  <Name>family</Name>
  <Value>Muntz</Value>
</Metadata>
<ContentLength>5</ContentLength>
<AccessControlList>
  <Grant>
    <Grantee xsi:type="CanonicalUser">
      <ID>a9a7b886d6241bf9b1c61be666e9</ID>
      <DisplayName>chriscustomer</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
  <Grant>
    <Grantee xsi:type="Group">
      <URI>http://acs.amazonaws.com/groups/global/AllUsers<URI>
    </Grantee>
    <Permission>READ</Permission>
  </Grant>
</AccessControlList>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2007-05-11T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObject>
```

#### Sample Response

```
<PutObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <PutObjectResponse>
    <ETag>"828ef3fd9a96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2006-03-01T12:00:00.183Z</LastModified>
  </PutObjectResponse>
</PutObjectResponse>
```

## Elements

- **Bucket**: The bucket in which to add the object.
- **Key**: The key to assign to the object.
- **Metadata**: You can provide name-value metadata pairs in the metadata element. These will be stored with the object.
- **ContentLength**: The length of the data in bytes.
- **AccessControlList**: An Access Control List for the resource. This element is optional. If omitted, the requester is given FULL\_CONTROL access to the object. If the object already exists, the preexisting Access Control Policy is replaced.

## Responses

- **ETag**: The entity tag is an MD5 hash of the object that you can use to do conditional fetches of the object using GetObjectExtended. The ETag only reflects changes to the contents of an object, not its metadata.
- **LastModified**: The Amazon S3 timestamp for the saved object.

## Access Control

To put objects into a bucket, you must have WRITE access to the bucket.

## Related Resources

- [CopyObject \(SOAP API\) \(p. 555\)](#)

## CopyObject (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

### Description

The `CopyObject` operation creates a copy of an object when you specify the key and bucket of a source object and the key and bucket of a target destination.

When copying an object, you can preserve all metadata (default) or specify new metadata. However, the ACL is not preserved and is set to `private` for the user making the request. To override the default ACL setting, specify a new ACL when generating a copy request. For more information, see [Using ACLs](#).

All copy requests must be authenticated. Additionally, you must have `read` access to the source object and `write` access to the destination bucket. For more information, see [Using Auth Access](#).

To only copy an object under certain conditions, such as whether the Etag matches or whether the object was modified before or after a specified date, use the request parameters `CopySourceIfUnmodifiedSince`, `CopyIfUnmodifiedSince`, `CopySourceIfMatch`, or `CopySourceIfNoneMatch`.

### Note

You might need to configure the SOAP stack socket timeout for copying large objects.

## Request Syntax

```
<CopyObject xmlns="http://bucket_name.s3.amazonaws.com/2006-03-01">
  <SourceBucket>source_bucket</SourceBucket>
  <SourceObject>source_object</SourceObject>
  <DestinationBucket>destination_bucket</DestinationBucket>
  <DestinationObject>destination_object</DestinationObject>
  <MetadataDirective>{REPLACE | COPY}</MetadataDirective>
  <Metadata>
    <Name>metadata_name</Name>
    <Value>metadata_value</Value>
  </Metadata>
  ...
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="user_type">
        <ID>user_id</ID>
        <DisplayName>display_name</DisplayName>
      </Grantee>
      <Permission>permission</Permission>
    </Grant>
    ...
  </AccessControlList>
  <CopySourceIfMatch>etag</CopySourceIfMatch>
  <CopySourceIfNoneMatch>etag</CopySourceIfNoneMatch>
  <CopySourceIfModifiedSince>date_time</CopySourceIfModifiedSince>
  <CopySourceIfUnmodifiedSince>date_time</CopySourceIfUnmodifiedSince>
  <AWSAccessKeyId>AWSAccessKeyId</AWSAccessKeyId>
  <Timestamp>TimeStamp</Timestamp>
  <Signature>Signature</Signature>
```

```
</CopyObject>
```

## Request Parameters

Name	Description	Required
SourceBucket	<p>The name of the source bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: A valid source bucket.</p>	Yes
SourceKey	<p>The key name of the source object.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: The key for a valid source object to which you have READ access.</p>	Yes
DestinationBucket	<p>The name of the destination bucket.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: You must have WRITE access to the destination bucket.</p>	Yes
DestinationKey	<p>The key of the destination object.</p> <p>Type: String</p> <p>Default: None</p> <p>Constraints: You must have WRITE access to the destination bucket.</p>	Yes
MetadataDirective	<p>Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.</p> <p>Type: String</p> <p>Default: COPY</p> <p>Valid values: COPY   REPLACE</p> <p>Constraints: Values other than COPY or REPLACE will result in an immediate error. You cannot copy an object to itself unless the MetadataDirective header is specified and its value set to REPLACE.</p>	No
Metadata	Specifies metadata name-value pairs to set for the object. If MetadataDirective is set to COPY, all metadata is ignored.	No

Name	Description	Required
	Type: String  Default: None  Constraints: None.	
AccessControlList	Grants access to users by e-mail addresses or canonical user ID.  Type: String  Default: None  Constraints: None	No
CopySourceIfMatch	Copies the object if its entity tag (ETag) matches the specified tag; otherwise return a PreconditionFailed.  Type: String  Default: None  Constraints: None. If the Etag does not match, the object is not copied.	No
CopySourceIfNoneMatch	Copies the object if its entity tag (ETag) is different than the specified Etag; otherwise returns an error.  Type: String  Default: None  Constraints: None.	No
CopySourceIfUnmodifiedSince	Copies the object if it hasn't been modified since the specified time; otherwise returns a PreconditionFailed.  Type: dateTime  Default: None	No
CopySourceIfModifiedSince	Copies the object if it has been modified since the specified time; otherwise returns an error.  Type: dateTime  Default: None	No

## Response Syntax

```
<CopyObjectResponse xmlns="http://bucket_name.s3.amazonaws.com/2006-03-01">
<CopyObjectResponse>
  <ETag>"etag"</ETag>
  <LastModified>timestamp</LastModified>
```

```
</CopyObjectResponse>
</CopyObjectResponse>
```

## Response Elements

Following is a list of response elements.

### Note

The SOAP API does not return extra whitespace. Extra whitespace is only returned by the REST API.

Name	Description
Etag	Returns the etag of the new object. The ETag only reflects changes to the contents of an object, not its metadata.  Type: String  Ancestor: CopyObjectResult
LastModified	Returns the date the object was last modified.  Type: String  Ancestor: CopyObjectResult

For information about general response elements, see [Using REST Error Response Headers](#).

## Special Errors

There are no special errors for this operation. For information about general Amazon S3 errors, see [List of Error Codes \(p. 7\)](#).

## Examples

This example copies the `flotsam` object from the `pacific` bucket to the `jetsam` object of the `atlantic` bucket, preserving its metadata.

### Sample Request

```
<CopyObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <SourceBucket>pacific</SourceBucket>
  <SourceObject>flotsam</SourceObject>
  <DestinationBucket>atlantic</DestinationBucket>
  <DestinationObject>jetsam</DestinationObject>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzfq7RrtSFmw=</Signature>
</CopyObject>
```

### Sample Response

```
<CopyObjectResponse xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <CopyObjectResponse>
    <ETag>"828ef3fdfa96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2008-02-18T13:54:10.183Z</LastModified>
  </CopyObjectResponse>
```

```
</CopyObjectResponse>
```

This example copies the "tweedledee" object from the wonderland bucket to the "tweedledum" object of the wonderland bucket, replacing its metadata.

### Sample Request

```
<CopyObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<SourceBucket>wonderland</SourceBucket>
<SourceObject>tweedledee</SourceObject>
<DestinationBucket>wonderland</DestinationBucket>
<DestinationObject>tweedledum</DestinationObject>
<MetadataDirective>REPLACE</MetadataDirective>
<Metadata>
  <Name>Content-Type</Name>
  <Value>text/plain</Value>
</Metadata>
<Metadata>
  <Name>relationship</Name>
  <Value>twins</Value>
</Metadata>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</CopyObject>
```

### Sample Response

```
<CopyObjectResponse xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<CopyObjectResponse>
  <ETag>"828ef3fd9a96f00ad9f27c383fc9ac7f"</ETag>
  <LastModified>2008-02-18T13:54:10.183Z</LastModified>
</CopyObjectResponse>
</CopyObjectResponse>
```

## Related Resources

- [PutObject \(SOAP API\) \(p. 553\)](#)
- [PutObjectInline \(SOAP API\) \(p. 551\)](#)

## GetObject (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The GetObject operation returns the current version of an object. If you try to GetObject an object that has a delete marker as its current version, S3 returns a 404 error. You cannot use the SOAP API to retrieve a specified version of an object. To do that, use the REST API. For more information, see [Versioning](#). For more options, use the [GetObjectExtended \(SOAP API\) \(p. 563\)](#) operation.

### Example

This example gets the "Nelson" object from the "quotes" bucket.

### Sample Request

```
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
```

```
<Bucket>quotes</Bucket>
<Key>Nelson</Key>
<GetMetadata>true</GetMetadata>
<GetData>true</GetData>
<InlineData>true</InlineData>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

#### Sample Response

```
<GetObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>aGEtaGE=</Data>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fd9a96f00ad9f27c383fc9ac7f"</ETag>
  </GetObjectResponse>
</GetObjectResponse>
```

## Elements

- **Bucket**: The bucket from which to retrieve the object.
- **Key**: The key that identifies the object.
- **GetMetadata**: The metadata is returned with the object if this is true.
- **GetData**: The object data is returned if this is true.
- **InlineData**: If this is true, then the data is returned, base 64-encoded, as part of the SOAP body of the response. If false, then the data is returned as a SOAP attachment. The **InlineData** option is not suitable for use with large objects. The system limits this operation to working with 1MB of data or less. A **GetObject** request with the **InlineData** flag set will fail with the **InlineDataTooLargeError** status code if the resulting Data parameter would have encoded more than 1MB. To download large objects, consider calling **GetObject** without setting the **InlineData** flag, or use the REST API instead.

## Returned Elements

- **Metadata**: The name-value paired metadata stored with the object.
- **Data**: If **InlineData** was true in the request, this contains the base 64 encoded object data.
- **LastModified**: The time that the object was stored in Amazon S3.
- **ETag**: The object's entity tag. This is a hash of the object that can be used to do conditional gets. The ETag only reflects changes to the contents of an object, not its metadata.

## Access Control

You can read an object only if you have been granted **READ** access to the object.

## SOAP Chunked and Resumable Downloads

To provide GET flexibility, Amazon S3 supports chunked and resumable downloads.

Select from the following:

- For large object downloads, you might want to break them into smaller chunks. For more information, see [Range GETs \(p. 561\)](#)
- For GET operations that fail, you can design your application to download the remainder instead of the entire file. For more information, see [REST GET Error Recovery \(p. 563\)](#)

### Range GETs

For some clients, you might want to break large downloads into smaller downloads. To break a GET into smaller units, use Range.

Before you can break a GET into smaller units, you must determine its size. For example, the following request gets the size of the bigfile object.

```
<ListBucket xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<Bucket>bigbucket</Bucket>
<Prefix>bigfile</Prefix>
<MaxKeys>1</MaxKeys>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

Amazon S3 returns the following response.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01">
<Name>quotes</Name>
<Prefix>N</Prefix>
<MaxKeys>1</MaxKeys>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>bigfile</Key>
<LastModified>2006-01-01T12:00:00.000Z</LastModified>
<ETag>"828ef3fdfa96f00ad9f27c383fc9ac7f"</ETag>
<Size>2023276</Size>
<StorageClass>STANDARD</StorageClass>
<Owner>
<ID>bcaf1ffd86f41161ca5fb16fd081034f</ID>
<DisplayName>bigfile</DisplayName>
</Owner>
</Contents>
</ListBucketResult>
```

Following is a request that downloads the first megabyte from the bigfile object.

```
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<Bucket>bigbucket</Bucket>
<Key>bigfile</Key>
<GetMetadata>true</GetMetadata>
<GetData>true</GetData>
<InlineData>true</InlineData>
<ByteRangeStart>0</ByteRangeStart>
<ByteRangeEnd>1048576</ByteRangeEnd>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
```

```
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Amazon S3 returns the first megabyte of the file and the Etag of the file.

```
<GetObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>--first megabyte of bigfile--</Data>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>"828ef3fdfa96f00ad9f27c383fc9ac7f"</ETag>
  </GetObjectResponse>
</GetObjectResponse>
```

To ensure the file did not change since the previous portion was downloaded, specify the IfMatch element. Although the IfMatch element is not required, it is recommended for content that is likely to change.

The following is a request that gets the remainder of the file, using the IfMatch request header.

```
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>bigbucket</Bucket>
  <Key>bigfile</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <ByteRangeStart>10485761</ByteRangeStart>
  <ByteRangeEnd>2023276</ByteRangeEnd>
  <IfMatch>"828ef3fdfa96f00ad9f27c383fc9ac7f"</IfMatch>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

Amazon S3 returns the following response and the remainder of the file.

```
<GetObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <GetObjectResponse>
    <Status>
      <Code>200</Code>
      <Description>OK</Description>
    </Status>
    <Metadata>
      <Name>Content-Type</Name>
      <Value>text/plain</Value>
    </Metadata>
    <Metadata>
      <Name>family</Name>
      <Value>Muntz</Value>
    </Metadata>
    <Data>--remainder of bigfile--</Data>
```

```
<LastModified>2006-01-01T12:00:00.000Z</LastModified>
<ETag>"828ef3fd9a96f00ad9f27c383fc9ac7f"</ETag>
</GetObjectResponse>
</GetObjectResponse>
```

## Versioned GetObject

The following request returns the specified version of the object in the bucket.

```
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<Bucket>quotes</Bucket>
<Key>Nelson</Key>
<GetMetadata>true</GetMetadata>
<GetData>true</GetData>
<InlineData>true</InlineData>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObject>
```

## Sample Response

```
<GetObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
<GetObjectResponse>
<Status>
<Code>200</Code>
<Description>OK</Description>
</Status>
<Metadata>
<Name>Content-Type</Name>
<Value>text/plain</Value>
</Metadata>
<Metadata>
<Name>family</Name>
<Value>Muntz</Value>
</Metadata>
<Data>aGEtaGE=</Data>
<LastModified>2006-01-01T12:00:00.000Z</LastModified>
<ETag>"828ef3fd9a96f00ad9f27c383fc9ac7f"</ETag>
</GetObjectResponse>
</GetObjectResponse>
```

## REST GET Error Recovery

If an object GET fails, you can get the rest of the file by specifying the range to download. To do so, you must get the size of the object using [ListBucket](#) and perform a range GET on the remainder of the file. For more information, see [GetObjectExtended \(SOAP API\) \(p. 563\)](#).

## Related Resources

[Operations on Objects \(SOAP API\) \(p. 551\)](#)

## GetObjectExtended (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

GetObjectExtended is exactly like [GetObject \(SOAP API\) \(p. 559\)](#), except that it supports the following additional elements that can be used to accomplish much of the same functionality provided by HTTP GET headers (go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>).

GetObjectExtended supports the following elements in addition to those supported by GetObject:

- **ByteRangeStart**, **ByteRangeEnd**: These elements specify that only a portion of the object data should be retrieved. They follow the behavior of the HTTP byte ranges (go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35>).
- **IfModifiedSince**: Return the object only if the object's timestamp is later than the specified timestamp. (<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.25>)
- **IfUnmodifiedSince**: Return the object only if the object's timestamp is earlier than or equal to the specified timestamp. (go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.28>)
- **IfMatch**: Return the object only if its ETag matches the supplied tag(s). (go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.24>)
- **IfNoneMatch**: Return the object only if its ETag does not match the supplied tag(s). (go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.26>)
- **ReturnCompleteObjectOnConditionFailure**: ReturnCompleteObjectOnConditionFailure: If true, then if the request includes a range element and one or both of IfUnmodifiedSince/IfMatch elements, and the condition fails, return the entire object rather than a fault. This enables the If-Range functionality (go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.27>).

## DeleteObject (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The DeleteObject operation removes the specified object from Amazon S3. Once deleted, there is no method to restore or undelete an object.

### Note

If you delete an object that does not exist, Amazon S3 will return a success (not an error message).

### Example

This example deletes the "Nelson" object from the "quotes" bucket.

#### Sample Request

```
<DeleteObject xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AWSAccessKeyId> AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</DeleteObject>
```

#### Sample Response

```
<DeleteObjectResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <DeleteObjectResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </DeleteObjectResponse>
</DeleteObjectResponse>
```

## Elements

- **Bucket**: The bucket that holds the object.

- **Key:** The key that identifies the object.

## Access Control

You can delete an object only if you have **WRITE** access to the bucket, regardless of who owns the object or what rights are granted to it.

## GetObjectAccessControlPolicy (SOAP API)

### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `GetObjectAccessControlPolicy` operation fetches the access control policy for an object.

### Example

This example retrieves the access control policy for the "Nelson" object from the "quotes" bucket.

#### Sample Request

```
<GetObjectAccessControlPolicy xmlns="http://doc.s3.amazonaws.com/2006-03-01">
<Bucket>quotes</Bucket>
<Key>Nelson</Key>
<AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
<Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
<Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</GetObjectAccessControlPolicy>
```

#### Sample Response

```
<AccessControlPolicy>
<Owner>
<ID>a9a7b886d6fd24a541bf9b1c61be666e9</ID>
<DisplayName>chriscustomer</DisplayName>
</Owner>
<AccessControlList>
<Grant>
<Grantee xsi:type="CanonicalUser">
<ID>a9a7b841bf9b1c61be666e9</ID>
<DisplayName>chriscustomer</DisplayName>
</Grantee>
<Permission>FULL_CONTROL</Permission>
</Grant>
<Grant>
<Grantee xsi:type="Group">
<URI>http://acs.amazonaws.com/groups/global/AllUsers<URI>
</Grantee>
<Permission>READ</Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

## Response Body

The response contains the access control policy for the bucket. For an explanation of this response, [SOAP Access Policy](#).

## Access Control

You must have `READ_ACP` rights to the object in order to retrieve the access control policy for an object.

### SetObjectAccessControlPolicy (SOAP API)

#### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

The `SetObjectAccessControlPolicy` operation sets the access control policy for an existing object. If successful, the previous access control policy for the object is entirely replaced with the specified access control policy.

#### Example

This example gives the specified user (usually the owner) `FULL_CONTROL` access to the "Nelson" object from the "quotes" bucket.

#### Sample Request

```
<SetObjectAccessControlPolicy xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</SetObjectAccessControlPolicy>
```

#### Sample Response

```
<SetObjectAccessControlPolicyResponse xmlns="http://s3.amazonaws.com/doc/2006-03-01">
  <SetObjectAccessControlPolicyResponse>
    <Code>200</Code>
    <Description>OK</Description>
  </SetObjectAccessControlPolicyResponse>
</SetObjectAccessControlPolicyResponse>
```

## Access Control

You must have `WRITE_ACP` rights to the object in order to set the access control policy for a bucket.

### SOAP Error Responses

#### Note

SOAP support over HTTP is deprecated, but it is still available over HTTPS. New Amazon S3 features will not be supported for SOAP. We recommend that you use either the REST API or the AWS SDKs.

In SOAP, an error result is returned to the client as a SOAP fault, with the HTTP response code 500. If you do not receive a SOAP fault, then your request was successful. The Amazon S3 SOAP fault code is

comprised of a standard SOAP 1.1 fault code (either "Server" or "Client") concatenated with the Amazon S3-specific error code. For example: "Server.InternalError" or "Client.NoSuchBucket". The SOAP fault string element contains a generic, human readable error message in English. Finally, the SOAP fault detail element contains miscellaneous information relevant to the error.

For example, if you attempt to delete the object "Fred", which does not exist, the body of the SOAP response contains a "NoSuchKey" SOAP fault.

The following example shows a sample SOAP error response.

```
<soapenv:Body>
<soapenv:Fault>
<Faultcode>soapenv:Client.NoSuchKey</Faultcode>
<Faultstring>The specified key does not exist.</Faultstring>
<Detail>
  <Key>Fred</Key>
</Detail>
</soapenv:Fault>
</soapenv:Body>
```

The following table explains the SOAP error response elements

Name	Description
Detail	<p>Container for the key involved in the error</p> <p>Type: Container</p> <p>Ancestor: Body.Fault</p>
Fault	<p>Container for error information.</p> <p>Type: Container</p> <p>Ancestor: Body</p>
Faultcode	<p>The fault code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see <a href="#">List of Error Codes (p. 7)</a>.</p> <p>Type: String</p> <p>Ancestor: Body.Fault</p>
Faultstring	<p>The fault string contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the fault string.</p> <p>Type: String</p> <p>Ancestor: Body.Fault</p>
Key	<p>Identifies the key involved in the error</p> <p>Type: String</p> <p>Ancestor: Body.Fault</p>

# Appendix: Lifecycle Configuration APIs (Deprecated)

Bucket lifecycle configuration is updated to support filters based on object tags. That is, you can now specify a rule that specifies key name prefix, one or more object tags, or both to select a subset of objects to which the rule applies. The APIs have been updated accordingly. The following topics describes the prior version of the PUT and GET bucket lifecycle operations for backward compatibility.

## Topics

- [PUT Bucket lifecycle \(Deprecated\) \(p. 569\)](#)
- [GET Bucket lifecycle \(Deprecated\) \(p. 579\)](#)

# PUT Bucket lifecycle (Deprecated)

## Description

### Important

For an updated version of this API, see [PUT Bucket lifecycle \(p. 265\)](#). This version has been deprecated. Existing lifecycle configurations will work. For new lifecycle configurations, use the updated API.

Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration. For information about lifecycle configuration, see [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

## Permissions

By default, all Amazon S3 resources, including buckets, objects, and related subresources (for example, lifecycle configuration and website configuration) are private. Only the resource owner, the AWS account that created the resource, can access it. The resource owner can optionally grant access permissions to others by writing an access policy. For this operation, users must get the s3:PutLifecycleConfiguration permission.

You can also explicitly deny permissions. Explicit denial also supersedes any other permissions. If you want to prevent users or accounts from removing or deleting objects from your bucket, you must deny them permissions for the following actions:

- s3:DeleteObject
- s3:DeleteObjectVersion
- s3:PutLifecycleConfiguration

For more information about permissions, see [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
PUT /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
Content-MD5: MD5

Lifecycle configuration in the request body
```

For details about authorization strings, see [Authenticating Requests \(AWS Signature Version 4\) \(p. 14\)](#).

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

Name	Description	Required
Content-MD5	The base64-encoded 128-bit MD5 digest of the data. You must use this header as a message	Yes

Name	Description	Required
	<p>integrity check to verify that the request body was not corrupted in transit. For more information, see <a href="#">RFC 1864</a>.</p> <p>Type: String</p> <p>Default: None</p>	

## Request Body

In the request, you specify the lifecycle configuration in the request body. The lifecycle configuration is specified as XML. The following is an example of a basic lifecycle configuration. It specifies one rule. The `Prefix` in the rule identifies objects to which the rule applies. The rule also specifies two actions (`Transition` and `Expiration`). Each action specifies a timeline when Amazon S3 should perform the action. The `Status` indicates whether the rule is enabled or disabled.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <Transition>
      <Date>value</Date>
      <StorageClass>storage class</StorageClass>
    </Transition>
    <Expiration>
      <Days>value</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

If the state of your bucket is versioning-enabled or versioning-suspended, you can have many versions of the same object: one current version and zero or more noncurrent versions. The following lifecycle configuration specifies the actions (`NoncurrentVersionTransition`, `NoncurrentVersionExpiration`) that are specific to noncurrent object versions.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>key-prefix</Prefix>
    <Status>rule-status</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>value</NoncurrentDays>
      <StorageClass>storage class</StorageClass>
    </NoncurrentVersionTransition>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>value</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

You can use the multipart upload API to upload large objects in parts. For more information about multipart uploads, see [Multipart Upload Overview](#) in the *Amazon Simple Storage Service Developer Guide*. With lifecycle configuration, you can tell Amazon S3 to abort incomplete multipart uploads, which are identified by the key name prefix specified in the rule, if they don't complete within a specified number of days. When Amazon S3 aborts a multipart upload, it deletes all parts associated with the upload. This ensures that you don't have incomplete multipart uploads that have left parts stored in Amazon S3, so

you don't have to pay storage costs for them. The following is an example lifecycle configuration that specifies a rule with the AbortIncompleteMultipartUpload action. This action tells Amazon S3 to abort incomplete multipart uploads seven days after initiation.

```
<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>SomeKeyPrefix</Prefix>
    <Status>rule-status</Status>
    <AbortIncompleteMultipartUpload>
      <DaysAfterInitiation>7</DaysAfterInitiation>
    </AbortIncompleteMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

The following table describes the XML elements in the lifecycle configuration.

Name	Description	Required
AbortIncompleteMultipartUpload	Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.  Child: DaysAfterInitiation  Type: Container  Ancestor: Rule	Yes, if no other action is specified for the rule
Date	Date when you want Amazon S3 to take the action. For more information, see <a href="#">Lifecycle Rules: Based on a Specific Date</a> in the <i>Amazon Simple Storage Service Developer Guide</i> .  The date value must conform to ISO 8601 format. The time is always midnight UTC.  Type: String  Ancestor: Expiration or Transition	Yes, if Days and ExpiredObjectDeleteMarker are absent
Days	Specifies the number of days after object creation when the specific rule action takes effect.  Type: Nonnegative Integer when used with Transition, Positive Integer when used with Expiration  Ancestor: Expiration, Transition	Yes, if Date and ExpiredObjectDeleteMarker are absent
DaysAfterInitiation	Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload.  Type: Positive Integer  Ancestor: AbortIncompleteMultipartUpload	Yes, if a parent tag is specified

Name	Description	Required
Expiration	<p>This action specifies a period in an object's lifetime when Amazon S3 should take the appropriate expiration action. The action Amazon S3 takes depends on whether the bucket is versioning-enabled.</p> <ul style="list-style-type: none"> <li>If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</li> <li>If the bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. A versioning-enabled bucket can have many versions of the same object: one current version and zero or more noncurrent versions.</li> </ul> <p>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</p> <p><b>Important</b> If a bucket's state is versioning-suspended, Amazon S3 creates a delete marker with version ID <code>null</code>. If you have a version with version ID <code>null</code>, Amazon S3 overwrites that version.</p> <p><b>Note</b> To set the expiration for noncurrent objects, use the <code>NoncurrentVersionExpiration</code> action.</p> <p>Type: Container Children: Days or Date Ancestor: Rule</p>	Yes, if no other action is present in the Rule.
ID	<p>Unique identifier for the rule. The value cannot be longer than 255 characters.</p> <p>Type: String Ancestor: Rule</p>	No
LifecycleConfiguration	<p>Container for lifecycle rules. You can add as many as 1000 rules.</p> <p>Type: Container Children: Rule Ancestor: None</p>	Yes

Name	Description	Required
ExpiredObjectDeleteMarker	<p>On a versioned bucket (a versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to tell Amazon S3 to delete expired object delete markers. For an example, see <a href="#">Example 8: Removing Expired Object Delete Markers</a> in the <i>Amazon Simple Storage Service Developer Guide</i>. Don't add it to a non-versioned bucket, because that type of bucket cannot include delete markers.</p> <p>Type: String</p> <p>Valid values: true   false (the value false is allowed, but it is no-op, which means that Amazon S3 will not take action)</p> <p>Ancestor: <a href="#">Expiration</a></p>	Yes, if Date and Days are absent
NoncurrentDays	<p>Specifies the number of days an object is noncurrent before Amazon S3 can perform the associated action. For information about the noncurrent days calculations, see <a href="#">How Amazon S3 Calculates When an Object Became Noncurrent</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Nonnegative Integer when used with <a href="#">NoncurrentVersionTransition</a>, Positive Integer when used with <a href="#">NoncurrentVersionExpiration</a></p> <p>Ancestor: <a href="#">NoncurrentVersionExpiration</a> or <a href="#">NoncurrentVersionTransition</a></p>	Yes
NoncurrentVersionExpiration	<p>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.</p> <p>Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to tell Amazon S3 to delete noncurrent object versions at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: <a href="#">NoncurrentDays</a></p> <p>Ancestor: <a href="#">Rule</a></p>	Yes, if no other action is present in the Rule

Name	Description	Required
NoncurrentVersionTransition	<p>Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class.</p> <p>If your bucket is versioning-enabled (or if versioning is suspended), you can set this action to tell Amazon S3 to transition noncurrent object versions at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: NoncurrentDays and StorageClass</p> <p>Ancestor: Rule</p>	Yes, if no other action is present in the Rule
Prefix	<p>Object key prefix that identifies one or more objects to which the rule applies.</p> <p>Type: String</p> <p>Ancestor: Rule</p>	Yes
Rule	<p>Container for a lifecycle rule. A lifecycle configuration can contain as many as 1000 rules.</p> <p>Type: Container</p> <p>Ancestor:LifecycleConfiguration</p>	Yes
Status	<p>If enabled, Amazon S3 executes the rule as scheduled. If it is disabled, Amazon S3 ignores the rule.</p> <p>Type: String</p> <p>Ancestor: Rule</p> <p>Valid values: Enabled, Disabled</p>	Yes
StorageClass	<p>Specifies the Amazon S3 storage class to which you want the object to transition.</p> <p>Type: String</p> <p>Ancestor: Transition and NoncurrentVersionTransition</p> <p>Valid values: STANDARD_IA   ONEZONE_IA   GLACIER</p>	Yes  This element is required only if you specify one or both its ancestors.

Name	Description	Required
Transition	<p>This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.</p> <ul style="list-style-type: none"> <li>If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.</li> <li>If your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of objects identified in the rule.</li> </ul> <p><b>Note</b> A versioning-enabled bucket can have many versions of an object. This action has no effect on noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.</p> <p>Type: Container Children: Days or Date, and StorageClass Ancestor: Rule</p>	Yes, if no other action is present in the Rule

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of the operation does not return response elements.

### Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Add Lifecycle Configuration to a Bucket That Is Not Versioning-enabled

The following lifecycle configuration specifies two rules, each with one action.

- The Transition action tells Amazon S3 to transition objects with the "documents/" prefix to the GLACIER storage class 30 days after creation.
- The Expiration action tells Amazon S3 to delete objects with the "logs/" prefix 365 days after creation.

```
<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample `PUT /?lifecycle` request that adds the preceding lifecycle configuration to the `examplebucket` bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:11:21 GMT
Content-MD5: q6yJD1IkBaGGfb3QLY69A==
Authorization: authorization string
Content-Length: 415

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: r+qR7+nhXtJDDIJ0JJYcd+1j5nM/rUFiiiz/fNbDOsd3JUE8NWMLNHXmvPfwMpdc
x-amz-request-id: 9E26D08072A8EF9E
Date: Wed, 14 May 2014 02:11:22 GMT
```

```
Content-Length: 0
Server: AmazonS3
```

## Example 2: Add Lifecycle Configuration to a Versioning-enabled Bucket

The following lifecycle configuration specifies two rules, each with one action for Amazon S3 to perform. You specify these actions when your bucket is versioning-enabled or versioning is suspended:

- The `NoncurrentVersionExpiration` action tells Amazon S3 to expire noncurrent versions of objects with the "logs/" prefix 100 days after the objects become noncurrent.
- The `NoncurrentVersionTransition` action tells Amazon S3 to transition noncurrent versions of objects with the "documents/" prefix to the `GLACIER` storage class 30 days after they become noncurrent.

```
<LifeCycleConfiguration>
<Rule>
  <ID>DeleteAfterBecomingNonCurrent</ID>
  <Prefix>logs/</Prefix>
  <Status>Enabled</Status>
  <NoncurrentVersionExpiration>
    <NoncurrentDays>100</NoncurrentDays>
  </NoncurrentVersionExpiration>
</Rule>
<Rule>
  <ID>TransitionAfterBecomingNonCurrent</ID>
  <Prefix>documents/</Prefix>
  <Status>Enabled</Status>
  <NoncurrentVersionTransition>
    <NoncurrentDays>30</NoncurrentDays>
    <StorageClass>GLACIER</StorageClass>
  </NoncurrentVersionTransition>
</Rule>
</LifeCycleConfiguration>
```

The following is a sample `PUT /?lifecycle` request that adds the preceding lifecycle configuration to the `examplebucket` bucket.

```
PUT /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
x-amz-date: Wed, 14 May 2014 02:21:48 GMT
Content-MD5: 96rxH9mDqVNKkaZDddgnw==
Authorization: authorization string
Content-Length: 598

<LifeCycleConfiguration>
<Rule>
  <ID>DeleteAfterBecomingNonCurrent</ID>
  <Prefix>logs/</Prefix>
  <Status>Enabled</Status>
  <NoncurrentVersionExpiration>
    <NoncurrentDays>1</NoncurrentDays>
  </NoncurrentVersionExpiration>
</Rule>
<Rule>
  <ID>TransitionSoonAfterBecomingNonCurrent</ID>
  <Prefix>documents/</Prefix>
  <Status>Enabled</Status>
  <NoncurrentVersionTransition>
    <NoncurrentDays>0</NoncurrentDays>
    <StorageClass>GLACIER</StorageClass>
  </NoncurrentVersionTransition>
```

```
</Rule>
</LifeCycleConfiguration>
```

The following is a sample response.

```
HTTP/1.1 200 OK
x-amz-id-2: aXQ+KbIrmMmoO//3bMdDTw/CnjArwje+J49Hf+j44yRb/VmbIkgl05A+PT98Cp/6k07hf+LD2mY=
x-amz-request-id: 02D7EC4C10381EB1
Date: Wed, 14 May 2014 02:21:50 GMT
Content-Length: 0
Server: AmazonS3
```

## Additional Examples

For more examples of transitioning objects to storage classes such as STANDARD\_IA or ONEZONE\_IA, see [Examples of Lifecycle Configuration](#).

## Related Resources

- [GET Bucket lifecycle \(p. 145\)](#)
- [POST Object restore \(p. 397\)](#)
- By default, a resource owner—in this case, a bucket owner, which is the AWS account that created the bucket—can perform any of the operations. A resource owner can also grant others permission to perform the operation. For more information, see the following topics in the *Amazon Simple Storage Service Developer Guide*:
  - [Specifying Permissions in a Policy](#)
  - [Managing Access Permissions to Your Amazon S3 Resources](#)

# GET Bucket lifecycle (Deprecated)

## Description

### Important

For an updated version of this API, see [GET Bucket lifecycle \(p. 145\)](#). If you configured a bucket lifecycle using the <filter> element, you should see an updated version of this topic. This topic is provided for backward compatibility.

Returns the lifecycle configuration information set on the bucket. For information about lifecycle configuration, go to [Object Lifecycle Management](#) in the *Amazon Simple Storage Service Developer Guide*.

To use this operation, you must have permission to perform the s3:GetLifecycleConfiguration action. The bucket owner has this permission by default. The bucket owner can grant this permission to others. For more information about permissions, see [Managing Access Permissions to Your Amazon S3 Resources](#) in the *Amazon Simple Storage Service Developer Guide*.

## Requests

### Syntax

```
GET /?lifecycle HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: authorization string (see Authenticating Requests \(AWS Signature Version 4\))
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation uses only request headers that are common to all operations. For more information, see [Common Request Headers \(p. 2\)](#).

### Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

This implementation of the operation uses only response headers that are common to most responses. For more information, see [Common Response Headers \(p. 4\)](#).

### Response Elements

This implementation of GET returns the following response elements.

Name	Description	Required
AbortIncompleteMultipartUpload	Container for specifying when an incomplete multipart upload becomes eligible for an abort operation.	Yes, if no other action is specified for the rule

Name	Description	Required
	<p>Child: DaysAfterInitiation</p> <p>Type: Container</p> <p>Ancestor: Rule</p>	
Date	<p>Date when you want Amazon S3 to take the action. For more information, see <a href="#">Lifecycle Rules: Based on a Specific Date</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>The date value must conform to the ISO 8601 format. The time is always midnight UTC.</p> <p>Type: String</p> <p>Ancestor: Expiration or Transition</p>	Yes, if Days and ExpiredObjectDeleteMarker are absent
Days	<p>Specifies the number of days after object creation when the specific rule action takes effect. The object's eligibility time is calculated as creation time + the number of days with the resulting time rounded to midnight UTC of the next day.</p> <p>Type: Non-negative Integer when used with Transition, Positive Integer when used with Expiration.</p> <p>Ancestor: Transition or Expiration</p>	Yes, if Date and ExpiredObjectDeleteMarker are absent
DaysAfterInitiation	<p>Specifies the number of days after initiating a multipart upload when the multipart upload must be completed. If it does not complete by the specified number of days, it becomes eligible for an abort operation and Amazon S3 aborts the incomplete multipart upload.</p> <p>Type: Positive Integer</p> <p>Ancestor: AbortIncompleteMultipartUpload</p>	Yes, if Date is absent

Name	Description	Required
Expiration	<p>This action specifies a period in the object's lifetime when Amazon S3 should take the appropriate expiration action. The expiration action occurs only on objects that are eligible according to the period specified in the child Date or Days element. The action Amazon S3 takes depends on whether the bucket is versioning enabled.</p> <ul style="list-style-type: none"> <li>• If versioning has never been enabled on the bucket, Amazon S3 deletes the only copy of the object permanently.</li> <li>• Otherwise, if your bucket is versioning-enabled (or versioning is suspended), the action applies only to the current version of the object. Buckets that are versioning-enabled or versioning-suspended can have many versions of the same object: one current version, and zero or more noncurrent versions.</li> </ul> <p>Instead of deleting the current version, Amazon S3 makes it a noncurrent version by adding a delete marker as the new current version.</p> <p><b>Important</b> If the state of a bucket is versioning-suspended, Amazon S3 creates a delete marker with version ID null. If you have a version with version ID null, then Amazon S3 overwrites that version.</p> <p><b>Note</b> To set the expiration for noncurrent objects, you must use the NoncurrentVersionExpiration action.</p> <p>Type: Container Children: Days or Date Ancestor: Rule</p>	Yes, if the parent tag is specified
ID	<p>Unique identifier for the rule. The value cannot be longer than 255 characters.</p> <p>Type: String Ancestor: Rule</p>	No

Name	Description	Required
LifecycleConfiguration	<p>Container for lifecycle rules. You can add as many as 1000 rules.</p> <p>Type: Container</p> <p>Children: Rule</p> <p>Ancestor: None</p>	Yes
ExpiredObjectDeleteMarker	<p>On a versioned bucket (versioning-enabled or versioning-suspended bucket), this element indicates whether Amazon S3 will delete any expired object delete markers in the bucket. For an example, go to <a href="#">Example 8: Specify Expiration Action to Remove Expired Object Delete Markers</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: String</p> <p>Valid values: true   false (the value false is allowed but it is no-op, Amazon S3 doesn't take action if the value is false)</p> <p>Ancestor: Expiration</p>	Yes, if Date and Days are absent
NoncurrentDays	<p>Specifies the number of days that an object is noncurrent before Amazon S3 can perform the associated action. For information about calculating noncurrent days, see <a href="#">Lifecycle Rules Based on the Number of Days</a> in the <i>Amazon Simple Storage Service Developer Guide</i>.</p> <p>Type: Nonnegative Integer when used with NoncurrentVersionTransition, Positive Integer when used with NoncurrentVersionExpiration</p> <p>Ancestor: NoncurrentVersionExpiration or NoncurrentVersionTransition</p>	Yes, only if the ancestor is present
NoncurrentVersionExpiration	<p>Specifies when noncurrent object versions expire. Upon expiration, Amazon S3 permanently deletes the noncurrent object versions.</p> <p>Set this lifecycle configuration action on a bucket that has versioning enabled (or suspended) to request that Amazon S3 delete noncurrent object versions at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: NoncurrentDays</p> <p>Ancestor: Rule</p>	Yes, if no other action is present in the Rule

Name	Description	Required
NoncurrentVersionTransition	<p>Container for the transition rule that describes when noncurrent objects transition to the STANDARD_IA, ONEZONE_IA, or the GLACIER storage class.</p> <p>If your bucket is versioning-enabled (or versioning is suspended), you can set this action to request Amazon S3 to transition noncurrent object versions to the GLACIER storage class at a specific period in the object's lifetime.</p> <p>Type: Container</p> <p>Children: NoncurrentDays and StorageClass</p> <p>Ancestor: Rule</p>	Yes, if no other action is present in the Rule
Prefix	<p>Object key prefix identifying one or more objects to which the rule applies.</p> <p>Type: String</p> <p>Ancestor: Rule</p>	Yes
Rule	<p>Container for a lifecycle rule.</p> <p>Type: Container</p> <p>Ancestor: LifecycleConfiguration</p>	Yes
Status	<p>If Enabled, Amazon S3 executes the rule as scheduled. If Disabled, Amazon S3 ignores the rule.</p> <p>Type: String</p> <p>Ancestor: Rule</p> <p>Valid values: Enabled or Disabled</p>	Yes
StorageClass	<p>Specifies the Amazon S3 storage class to which you want to transition the object.</p> <p>Type: String</p> <p>Ancestor: Transition and NoncurrentVersionTransition</p> <p>Valid values: STANDARD_IA   ONEZONE_IA   GLACIER</p>	Yes

Name	Description	Required
Transition	<p>This action specifies a period in the objects' lifetime when Amazon S3 should transition them to the STANDARD_IA, ONEZONE_IA, or GLACIER storage class. When this action is in effect, what Amazon S3 does depends on whether the bucket is versioning-enabled.</p> <ul style="list-style-type: none"> <li>If versioning has never been enabled on the bucket, Amazon S3 transitions the only copy of the object to the specified storage class.</li> <li>When your bucket is versioning-enabled (or versioning is suspended), Amazon S3 transitions only the current versions of the objects identified in the rule.</li> </ul> <p><b>Note</b>  A versioning-enabled or versioning-suspended bucket can contain many versions of an object. This action has no effect on the noncurrent object versions. To transition noncurrent objects, you must use the NoncurrentVersionTransition action.</p> <p>Type: Container  Children: Days or Date, and StorageClass  Ancestor: Rule</p>	Yes, if no other action is present in the Rule

## Special Errors

Error Code	Description	HTTP Status Code	SOAP Fault Code Prefix
NoSuchLifecycleConfiguration	The lifecycle configuration does not exist.	404 Not Found	Client

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 6\)](#).

## Examples

### Example 1: Retrieve a Lifecycle Subresource

This example is a GET request to retrieve the lifecycle subresource from the specified bucket, and an example response with the returned lifecycle configuration.

#### Sample Request

```
GET /?lifecycle HTTP/1.1
Host: examplebucket.s3.amazonaws.com
```

```
x-amz-date: Thu, 15 Nov 2012 00:17:21 GMT
Authorization: signatureValue
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ITnGT1y4RyTmXa3rPi4hkLTYouTf0hccUjo0iCPjz6FnfIutBj3M7fPGlWO2SEWp
x-amz-request-id: 51991C342C575321
Date: Thu, 15 Nov 2012 00:17:23 GMT
Server: AmazonS3
Content-Length: 358

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Rule>
        <ID>Archive and then delete rule</ID>
        <Prefix>projectdocs/</Prefix>
        <Status>Enabled</Status>
        <Transition>
            <Days>30</Days>
            <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <Transition>
            <Days>365</Days>
            <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
            <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

## Related Resources

- [PUT Bucket lifecycle \(p. 265\)](#)
- [DELETE Bucket lifecycle \(p. 88\)](#)

# Glossary

---

100-continue	A method that enables a client to see if a server can accept a request before actually sending it. For large PUTs, this can save both time and bandwidth charges.
account	AWS account associated with a particular developer.
authentication	The process of proving your identity to the system.
bucket	A container for objects stored in Amazon S3. Every object is contained within a bucket. For example, if the object named photos/puppy.jpg is stored in the johnsmith bucket, then it is addressable using the URL <a href="http://johnsmith.s3.amazonaws.com/photos/puppy.jpg">http://johnsmith.s3.amazonaws.com/photos/puppy.jpg</a>
canned access policy	A standard access control policy that you can apply to a bucket or object. Valid Values: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control
canonicalization	The process of converting data into a standard format that will be recognized by a service such as Amazon S3.
consistency model	The method through which Amazon S3 achieves high availability, which involves replicating data across multiple servers within Amazon's data centers. After a "success" is returned, your data is safely stored. However, information about the changes might not immediately replicate across Amazon S3.
key	The unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Since a bucket and key together uniquely identify each object, Amazon S3 can be thought of as a basic data map between "bucket + key" and the object itself. Every object in Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, and key, as in <a href="http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl">http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl</a> , where "doc" is the name of the bucket, and "2006-03-01/AmazonS3.wsdl" is the key.
metadata	The metadata is a set of name-value pairs that describe the object. These include default metadata such as the date last modified and standard HTTP metadata such as Content-Type. The developer can also specify custom metadata at the time the Object is stored.
object	The fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3.

part	The fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3.
service endpoint	The host and port with which you are trying to communicate within the destination URL. For virtual hosted-style requests, this is <code>mybucket.s3.amazonaws.com</code> . For path-style requests, this is <code>s3.amazonaws.com</code>