

---

# **Amazon Simple Storage Service**

## **API Reference**

**API Version 2006-03-01**



## Amazon Simple Storage Service: API Reference

Copyright © 2010 Amazon Web Services LLC or its affiliates. All rights reserved.

## Table of Contents

|  |     |
|--|-----|
| Welcome to Amazon S3 .....                 | 1   |
| Amazon S3 API Reference Introduction ..... | 2   |
| Error Responses .....                      | 3   |
| REST API .....                             | 11  |
| Common Request Headers .....               | 12  |
| Common Response Headers .....              | 14  |
| Operations on the Service .....            | 15  |
| GET Service .....                          | 15  |
| Operations on Buckets .....                | 18  |
| DELETE Bucket .....                        | 19  |
| DELETE Bucket policy .....                 | 21  |
| DELETE Bucket website .....                | 23  |
| GET Bucket (List Objects) .....            | 25  |
| GET Bucket acl .....                       | 32  |
| GET Bucket policy .....                    | 35  |
| GET Bucket location .....                  | 37  |
| GET Bucket logging .....                   | 39  |
| GET Bucket notification .....              | 42  |
| GET Bucket Object versions .....           | 45  |
| GET Bucket requestPayment .....            | 57  |
| GET Bucket versioning .....                | 59  |
| GET Bucket website .....                   | 62  |
| List Multipart Uploads .....               | 65  |
| PUT Bucket .....                           | 74  |
| PUT Bucket acl .....                       | 78  |
| PUT Bucket policy .....                    | 82  |
| PUT Bucket logging .....                   | 84  |
| PUT Bucket notification .....              | 89  |
| PUT Bucket requestPayment .....            | 93  |
| PUT Bucket versioning .....                | 95  |
| PUT Bucket website .....                   | 99  |
| Operations on Objects .....                | 101 |
| DELETE Object .....                        | 102 |
| GET Object .....                           | 106 |
| GET Object ACL .....                       | 112 |
| Versioning .....                           | 112 |
| GET Object torrent .....                   | 116 |
| HEAD Object .....                          | 118 |
| Versioning .....                           | 118 |
| POST Object .....                          | 122 |
| Versioning .....                           | 122 |
| PUT Object .....                           | 129 |
| Versioning .....                           | 129 |
| PUT Object acl .....                       | 135 |
| PUT Object - Copy .....                    | 142 |
| Versioning .....                           | 142 |
| Initiate Multipart Upload .....            | 149 |
| Upload Part .....                          | 153 |
| Upload Part - Copy .....                   | 156 |
| Complete Multipart Upload .....            | 161 |
| Abort Multipart Upload .....               | 166 |
| List Parts .....                           | 168 |
| SOAP API .....                             | 173 |
| Operations on the Service .....            | 173 |
| ListAllMyBuckets .....                     | 173 |

|                                    |     |
|------------------------------------|-----|
| Operations on Buckets .....        | 174 |
| CreateBucket .....                 | 175 |
| DeleteBucket .....                 | 176 |
| ListBucket .....                   | 176 |
| GetBucketAccessControlPolicy ..... | 179 |
| SetBucketAccessControlPolicy ..... | 180 |
| GetBucketLoggingStatus .....       | 181 |
| SetBucketLoggingStatus .....       | 182 |
| Operations on Objects .....        | 183 |
| PutObjectInline .....              | 184 |
| PutObject .....                    | 186 |
| CopyObject .....                   | 189 |
| GetObject .....                    | 193 |
| GetObjectExtended .....            | 198 |
| DeleteObject .....                 | 198 |
| GetObjectAccessControlPolicy ..... | 199 |
| SetObjectAccessControlPolicy ..... | 200 |
| Amazon S3 Resources .....          | 202 |
| Document History .....             | 204 |
| Glossary .....                     | 207 |
| Index .....                        | 209 |

---

# Welcome to Amazon S3

---

This is the *Amazon Simple Storage Service (Amazon S3) API Reference*. It explains the Amazon S3 API interface. It describes various API operations, related request and response structures, and error codes.

Amazon Simple Storage Service (Amazon S3) is a web service that enables you to store data in the cloud. You can then download the data or use the data with other AWS services, such as Amazon Elastic Cloud Computer (see [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) ).

## How Do I...?

| Information                            | Relevant Sections   |
|--|---|
| General product overview and pricing   | <a href="#">Amazon Simple Storage Service (Amazon S3)</a> |
| List of REST Operations                | <a href="#">REST API (p. 11)</a>                          |
| List of SOAP Operations                | <a href="#">SOAP API (p. 173)</a>                         |
| Amazon S3 Error codes and descriptions | <a href="#">List of Error Codes (p. 3)</a>                |

# Amazon S3 API Reference

## Introduction

---

This application programming interface reference explains Amazon S3 operations, their parameters, responses, and errors. There are separate sections for the REST and SOAP APIs, which include example requests and responses.

The location of the latest Amazon S3 WSDL is <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>.

# Error Responses

---

This section provides reference information about Amazon S3 errors.

## List of Error Codes

The following table lists Amazon S3 error codes.

| Error Code                   | Description  | HTTP Status Code | SOAP Fault Code Prefix |
|------------------------------|--|------------------|------------------------|
| AccessDenied                 | Access Denied  | 403 Forbidden    | Client                 |
| AccountProblem               | There is a problem with your AWS account that prevents the operation from completing successfully. Please use <a href="#">Contact Us</a> .           | 403 Forbidden    | Client                 |
| AmbiguousGrantByEmailAddress | The e-mail address you provided is associated with more than one account.  | 400 Bad Request  | Client                 |
| 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      | Your previous request to create the named bucket succeeded and you already own it.   | 409 Conflict     | Client                 |
| BucketNotEmpty               | The bucket you tried to delete is not empty.   | 409 Conflict     | Client                 |

**Amazon Simple Storage Service API Reference**  
**List of Error Codes**

---

| Error Code                              | Description   | HTTP Status Code          | SOAP Fault Code Prefix |
|---|---|---------------------------|------------------------|
| 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                 |
| InlineDataTooLarge                      | Inline data exceeds the maximum allowed size.   | 400 Bad Request           | Client                 |
| InternalServerError                     | 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                 |
| InvalidDigest                           | The Content-MD5 you specified was an invalid.   | 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                 |



**Amazon Simple Storage Service API Reference**  
**List of Error Codes**

---

| Error Code                    | Description   | HTTP Status Code                    | SOAP Fault Code Prefix |
|-------------------------------|---|-------------------------------------|------------------------|
| 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.  | 403 Forbidden                       | Client                 |
| InvalidPolicyDocument         | The content of the form does not meet the conditions specified in the policy document.  | 400 Bad Request                     | Client                 |
| InvalidRange                  | The requested range cannot be satisfied.  | 416 Requested Range Not Satisfiable | Client                 |
| InvalidRequest                | SOAP requests must be made over an HTTPS connection.  | 400 Bad Request                     | Client                 |
| 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                 |
| InvalidURI                    | Couldn't parse the specified URI.   | 400 Bad Request                     | Client                 |
| KeyTooLong                    | 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                 |

**Amazon Simple Storage Service API Reference**  
**List of Error Codes**

---

| Error Code                        | Description  | HTTP Status Code       | SOAP Fault Code Prefix |
|-----------------------------------|--|------------------------|------------------------|
| MalformedXML                      | This happens when the user sends a 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 was missing a required header.  | 400 Bad Request        | Client                 |
| NoLoggingStatusForKey             | There is no such thing as a logging status sub-resource for a key.   | 400 Bad Request        | Client                 |
| NoSuchBucket                      | The specified bucket does not exist.   | 404 Not Found          | Client                 |
| NoSuchKey                         | The specified key 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                 |

**Amazon Simple Storage Service API Reference**  
**List of Error Codes**

---

| Error Code                   | Description  | HTTP Status Code        | SOAP Fault Code Prefix |
|------------------------------|--|-------------------------|------------------------|
| 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="http://aws.amazon.com/s3">http://aws.amazon.com/s3</a>  | 403 Forbidden           | Client                 |
| NotSuchBucketPolicy          | The specified bucket does not have a bucket policy.  | 404 Not Found           | Client                 |
| OperationAborted             | A conflicting conditional operation is currently in progress against this resource. Please try again.  | 409 Conflict            | Client                 |
| PermanentRedirect            | The bucket you are attempting to access must be addressed using the specified endpoint. Please 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                 |
| RequestIsNotMultiPartContent | Bucket POST must be of the enclosure-type multipart/form-data.   | 400 Bad Request         | Client                 |
| 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                 |

| Error Code                      | Description  | HTTP Status Code        | SOAP Fault Code Prefix |
|---------------------------------|--|-------------------------|------------------------|
| ServiceUnavailable              | Please reduce your request rate.   | 503 Service Unavailable | Server                 |
| SlowDown                        | Please 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 e-mail 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, please check the order of the fields. | 400 Bad Request         | Client                 |

## REST Error Responses

When there is an error, the header information contains:

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

The body of 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  |
|------------------|--|
| <i>Code</i>      | 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. 3)</a> .<br>Type: String<br>Ancestor: Error   |
| <i>Error</i>     | Container for all error elements.<br>Type: Container<br>Ancestor: None   |
| <i>Message</i>   | 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 to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the error message.<br>Type: String<br>Ancestor: Error |
| <i>RequestId</i> | ID of the request associated with the error.<br>Type: String<br>Ancestor: Error  |
| <i>Resource</i>  | The bucket or object that is involved in the error.<br>Type: String<br>Ancestor: Error   |

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](#).

## SOAP Error Responses

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   |
|--------------------|---|
| <i>Detail</i>      | Container for the key involved in the error<br>Type: Container<br>Ancestor: Body.Fault  |
| <i>Fault</i>       | Container for error information.<br>Type: Container<br>Ancestor: Body   |
| <i>Faultcode</i>   | 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. 3)</a> .<br>Type: String<br>Ancestor: Body.Fault   |
| <i>Faultstring</i> | 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.<br>Type: String<br>Ancestor: Body.Fault |
| <i>Key</i>         | Identifies the key involved in the error<br>Type: String<br>Ancestor: Body.Fault  |

# REST API

---

## Topics

- [Common Request Headers \(p. 12\)](#)
- [Common Response Headers \(p. 14\)](#)
- [Operations on the Service \(p. 15\)](#)
- [Operations on Buckets \(p. 18\)](#)
- [Operations on Objects \(p. 101\)](#)

This section contains information specific to the Amazon S3 REST API.

The examples in this guide use the newer virtual hosted-style method for accessing buckets instead of the path-style. Although the path-style is still supported for legacy applications, we recommend using the virtual-hosted style where applicable. For more information, see [Working with Amazon S3 Buckets](#)

The following example is a virtual hosted-style request that deletes the `puppy.jpg` file from the `mybucket` bucket.

```
DELETE /puppy.jpg HTTP/1.1
User-Agent: dotnet
Host: mybucket.s3.amazonaws.com
Date: Tue, 15 Jan 2008 21:20:27 +0000
x-amz-date: Tue, 15 Jan 2008 21:20:27 +0000
Authorization: AWS OPN5J17HBGZHT7JJ3X82:k3nL7gH3+PadhTEVn5EXAMPLE
```

The following example is a path-style version of the same request.

```
DELETE /mybucket/puppy.jpg HTTP/1.1
User-Agent: dotnet
Host: s3.amazonaws.com
Date: Tue, 15 Jan 2008 21:20:27 +0000
x-amz-date: Tue, 15 Jan 2008 21:20:27 +0000
Authorization: AWS OPN5J17HBGZHT7JJ3X82:k3nL7gH3+PadhTEVn5EXAMPLE
```

## Common Request Headers

Amazon S3 REST requests include headers that contain basic information about the request. The following table describes headers that can be used by all Amazon S3 REST requests.

| Header Name    | Description  | Required    |
|----------------|--|-------------|
| Authorization  | The information required for request authentication.<br>Type: String<br>Default: None  | Yes         |
| Content-Length | Length of the message (without the headers) according to RFC 2616.<br>Type: String<br>Default: None<br>Condition: Required for PUTs and operations that load XML, such as logging and ACLs.  | Conditional |
| Content-Type   | The content type of the resource. Example:<br><code>text/plain</code><br>Type: String<br>Default: None   | No          |
| Date           | The current date and time according to the requester.<br>Example: <code>Wed, 01 Mar 2009 12:00:00 GMT</code><br>Type: String<br>Default: None  | Yes         |
| 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 <a href="#">Amazon Simple Storage Service Developer Guide</a> .<br>Type: String<br>Default: None<br>Condition: Required for HTTP 1.1 (most toolkits add this header automatically); optional for HTTP/1.0 requests. | Conditional |



| Header Name          | Description   | Required    |
|----------------------|---|-------------|
| x-amz-security-token | <p>This header can be used in the following scenarios:</p> <ul style="list-style-type: none"><li>• <b>Provide security tokens for Amazon DevPay operations</b>—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>• <b>Provide security token when using temporary security credentials</b>—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>Type: String<br/>Default: None<br/>Condition: Required for requests that use Amazon DevPay and requests that are signed using temporary security credentials.</p> | Conditional |

## Common Response Headers

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

| Name                | Description   |
|---------------------|---|
| Content-Length      | The length in bytes of the body in the response.<br>Type: String<br>Default: None   |
| Connection          | specifies whether the connection to the server is open or closed.<br>Type: Enum<br>Valid Values: open   close<br>Default: None  |
| Date                | The date and time Amazon S3 responded, for example, Wed, 01 Mar 2009 12:00:00 GMT.<br>Type: String<br>Default: None   |
| ETag                | The entity tag is a hash of the object. The ETag only reflects changes to the contents of an object, not its metadata. The ETag is determined when an object is created. For objects created by the PUT Object operation and the POST Object operation, the ETag is a quoted, 32-digit hexadecimal string representing the MD5 digest of the object data. For other objects, the ETag may or may not be an MD5 digest of the object data. If the ETag is not an MD5 digest of the object data, it will contain one or more non-hexadecimal characters and/or will consist of less than 32 or more than 32 hexadecimal digits.<br>Type: String |
| Server              | The name of the server that created the response.<br>Type: String<br>Default: AmazonS3  |
| x-amz-delete-marker | Specifies whether the object returned was (true) or was not (false) a Delete Marker.<br>Type: Boolean<br>Valid Values: true   false<br>Default: false   |
| x-amz-id-2          | A special token that helps AWS troubleshoot problems.<br>Type: String<br>Default: None  |
| x-amz-request-id    | A value created by Amazon S3 that uniquely identifies the request. In the unlikely event that you have problems with Amazon S3, AWS can use this value to troubleshoot the problem.<br>Type: String<br>Default: None  |

| Name             | Description   |
|------------------|---|
| 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 <code>null</code>.</p> <p>Type: String</p> <p>Valid Values: <code>null</code>   any URL-ready, UTF-8 encoded string</p> <p>Default: <code>null</code></p> |

## Operations on the Service

### Topics

- [GET Service \(p. 15\)](#)

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

## 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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### Response Elements

| Name                          | Description   |
|-------------------------------|---|
| <i>Bucket</i>                 | Container for bucket information.<br>Type: Container<br>Children: Name, CreationDate<br>Ancestor: ListAllMyBucketsResult.Buckets  |
| <i>Buckets</i>                | Container for one or more buckets.<br>Type: Container<br>Children: Bucket<br>Ancestor: ListAllMyBucketsResult   |
| <i>CreationDate</i>           | Date the bucket was created.<br>Type: date ( of the form yyyy-mm-ddThh:mm:ss.timezone, e.g., 2009-02-03T16:45:09.000Z)<br>Ancestor: ListAllMyBucketsResult.Buckets.Bucket |
| <i>DisplayName</i>            | Bucket owner's display name.<br>Type: String<br>Ancestor: ListAllMyBucketsResult.Owner  |
| <i>ID</i>                     | Bucket owner's user ID.<br>Type: String<br>Ancestor: ListAllMyBucketsResult.Owner   |
| <i>ListAllMyBucketsResult</i> | Container for response.<br>Type: Container<br>Children: Owner, Buckets<br>Ancestor: None  |
| <i>Name</i>                   | Bucket's name.<br>Type: String<br>Ancestor: ListAllMyBucketsResult.Buckets.Bucket   |
| <i>Owner</i>                  | Container for bucket owner information.<br>Type: Container<br>Ancestor: ListAllMyBucketsResult  |

### 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 \(p. 3\)](#).

## 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 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Owner>
    <ID>bcafl1ffd86f461ca5fb16fd081034f</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 \(List Objects\) \(p. 25\)](#)
- [GET Object \(p. 106\)](#)

# Operations on Buckets

## Topics

- [DELETE Bucket](#) (p. 19)
- [DELETE Bucket policy](#) (p. 21)
- [DELETE Bucket website](#) (p. 23)
- [GET Bucket \(List Objects\)](#) (p. 25)
- [GET Bucket acl](#) (p. 32)
- [GET Bucket policy](#) (p. 35)
- [GET Bucket location](#) (p. 37)
- [GET Bucket logging](#) (p. 39)
- [GET Bucket notification](#) (p. 42)
- [GET Bucket Object versions](#) (p. 45)
- [GET Bucket requestPayment](#) (p. 57)
- [GET Bucket versioning](#) (p. 59)
- [GET Bucket website](#) (p. 62)
- [List Multipart Uploads](#) (p. 65)
- [PUT Bucket](#) (p. 74)
- [PUT Bucket acl](#) (p. 78)
- [PUT Bucket policy](#) (p. 82)
- [PUT Bucket logging](#) (p. 84)
- [PUT Bucket notification](#) (p. 89)
- [PUT Bucket requestPayment](#) (p. 93)
- [PUT Bucket versioning](#) (p. 95)
- [PUT Bucket website](#) (p. 99)

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



### Note

For information about access policies, see [REST Access Policy](#) .

# DELETE Bucket

## Description

This implementation of the `DELETE` operation 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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### 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 \(p. 3\)](#).

## Examples

### Sample Request

This request deletes the bucket named "quotes".

```
DELETE / HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

## Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagkKdBX2z7I1/jGhDeJ3j6s80
x-amz-request-id: 32FE2CEB32F5EE25
Date: Wed, 01 Mar 2009 12:00:00 GMT
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 74\)](#)
- [DELETE Object \(p. 102\)](#)



# DELETE Bucket policy

## Description

This implementation of the `DELETE` operation uses the *policy* subresource to delete the policy on a specified bucket. To use the operation, you must have *DeletePolicy* permissions on the specified bucket and be the bucket owner.

If you do not have *DeletePolicy* permissions, Amazon S3 returns a 403 `Access Denied` error. If you have the correct permissions, but are not the bucket owner, Amazon S3 returns a 405 `Method Not Allowed` error. If the bucket doesn't have a policy, Amazon S3 returns a 204 `No Content` error. There are restrictions about who can create bucket policies and which objects in a bucket they can apply to. For more information, go to [Using Bucket Policies](#).

## Requests

### Syntax

```
DELETE /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### 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 \(p. 3\)](#).

## 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: AWS VGhpcyBSAMPLEeSB1bHZpbmc=
```

### 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 \(p. 74\)](#)
- [DELETE Object \(p. 102\)](#)

# 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 S3 Developer Guide*.

## Requests

### Syntax

```
DELETE /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### 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: AWS 1DBSMRRRRRWZ1NWYWAG2:acxI7sW0+ugzxhf2AtcqRLgy70B=
```

### Sample Response

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

## Related Resources

- [GET Bucket website \(p. 62\)](#)
- [PUT Bucket website \(p. 99\)](#)

# GET Bucket (List Objects)

## Description

This implementation of the `GET` operation returns some or all (up to 1000) 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.

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 \(p. 15\)](#).

## Requests

### Syntax

```
GET / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### 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 |
|------------------|--|----------|
| <i>delimiter</i> | A delimiter is a character you use to group keys. All keys that contain the same string between the <i>prefix</i> , if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, <i>CommonPrefixes</i> . If you don't specify the <i>prefix</i> parameter, then the substring starts at the beginning of the key. The keys that are grouped under <i>CommonPrefixes</i> result element are not returned elsewhere in the response.<br>Type: String<br>Default: None | No       |
| <i>marker</i>    | Specifies the key to start with when listing objects in a bucket. Amazon S3 lists objects in alphabetical order.<br>Type: String<br>Default: None  | No       |
| <i>max-keys</i>  | Sets the maximum number of keys returned in the response body. The response might contain fewer keys but will never contain more. If there are additional keys that satisfy the search criteria but were not returned because <i>max-keys</i> was exceeded, the response contains <code>&lt;IsTruncated&gt;true&lt;/IsTruncated&gt;</code> . To return the additional keys, see <i>marker</i> .<br>Type: String<br>Default: 1000   | No       |

| Parameter     | Description   | Required |
|---------------|---|----------|
| <i>prefix</i> | 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 <i>prefix</i> to make groups in the same way you'd use a folder in a file system.)<br>Type: String<br>Default: None | No       |

## Request Elements

This implementation of the operation does not use request elements.

## Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

## 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 \(p. 14\)](#).

### Response Elements

| Name           | Description   |
|----------------|---|
| Contents       | Metadata about each object returned.<br>Type: XML metadata<br>Ancestor: ListBucketResult  |
| CommonPrefixes | A response can contain <i>CommonPrefixes</i> only if you specify a <i>delimiter</i> . When you do, <i>CommonPrefixes</i> contains all (if there are any) keys between <i>Prefix</i> and the next occurrence of the string specified by <i>delimiter</i> . In effect, <i>CommonPrefixes</i> lists keys that act like subdirectories in the directory specified by <i>Prefix</i> . For example, if <i>prefix</i> is <i>notes/</i> and <i>delimiter</i> is a slash (/), in <i>notes/summer/july</i> , the common prefix is <i>notes/summer/</i> . All of the keys rolled up in a common prefix count as a single return when calculating the number of returns. See <i>MaxKeys</i> .<br>Type: String<br>Ancestor: ListBucketResult |
| Delimiter      | 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 <i>CommonPrefixes</i> collection. These rolled-up keys are not returned elsewhere in the response. Each rolled up result counts as only one return against the <i>MaxKeys</i> value.<br>Type: String<br>Ancestor: ListBucketResult  |

**Amazon Simple Storage Service API Reference**  
**GET Bucket (List Objects)**

---

| Name         | Description   |
|--------------|---|
| DisplayName  | Object owner's name.<br>Type: String<br>Ancestor: ListBucketResult.Contents.Owner   |
| ETag         | The entity tag is an MD5 hash of the object. The ETag only reflects changes to the contents of an object, not its metadata.<br>Type: String<br>Ancestor: ListBucketResult.Contents  |
| ID           | Object owner's ID.<br>Type: Boolean<br>Ancestor: ListBucketResult.Contents.Owner  |
| IsTruncated  | Specifies whether ( <i>true</i> ) or not ( <i>false</i> ) all of the results were returned. All of the results may not be returned if the number of results exceeds that specified by <i>MaxKeys</i> .<br>Type: String<br>Ancestor: boolean |
| Key          | The object's key.<br>Type: String<br>Ancestor: ListBucketResult.Contents  |
| LastModified | Date and time the object was last modified.<br>Type: Date<br>Ancestor: ListBucketResult.Contents  |
| Marker       | Indicates where in the bucket to begin listing.<br>Type: String<br>Ancestor: ListBucketResult   |
| MaxKeys      | The maximum number of keys returned in the response body.<br>Type: String<br>Ancestor: ListBucketResult   |
| Name         | Name of the bucket.<br>Type: String<br>Ancestor: ListBucketResult   |
| Owner        | Bucket owner.<br>Type: String<br>Children: DisplayName, ID<br>Ancestor: ListBucketResult.Contents   CommonPrefixes  |
| Prefix       | Keys that begin with the indicated prefix.<br>Type: String<br>Ancestor: ListBucketResult  |

| Name         | Description   |
|--------------|---|
| Size         | Size in bytes of the object.<br>Type: String<br>Ancestor: ListBucketResult.Contents |
| StorageClass | Always STANDARD.<br>Type: String<br>Ancestor: ListBucketResult.Contents             |

## 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. 3).

## 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
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>8a6925ce4a7f21c32aa379004fef</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</StorageClass>
```



```
<Owner>
  <ID>8a69b1ddee97f21c32aa379004fef</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 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAXexeEHndCX/7GRnfTXxReKUQF28IfP
x-amz-request-id: 3B3C7C725673C630
Date: Wed, 01 Mar 2009 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>&quot;828ef3fd9fa96f00ad9f27c383fc9ac7f&quot;</ETag>
    <Size>5</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcafl61ca5fb16fd081034f</ID>
      <DisplayName>webfile</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>Neo</Key>
    <LastModified>2006-01-01T12:00:00.000Z</LastModified>
    <ETag>&quot;828ef3fd9fa96f00ad9f27c383fc9ac7f&quot;</ETag>
    <Size>4</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaflffd86a5fb16fd081034f</ID>
      <DisplayName>webfile</DisplayName>
```

```
</Owner>
</Contents>
</ListBucketResult>
```

## Sample Request Using Prefix and Delimiter

Assume 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 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

The key `sample.html` 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 return 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.html</Key>
    <LastModified>2011-02-26T01:56:20.000Z</LastModified>
    <ETag>"bfl1d737a4d46a19f3bced6905cc8b902"</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 value `"/`, and the `prefix` parameter with value `photos/2006/`.

```
GET ?prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

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 from the beginning of the key to 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/feb/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>photos/2006/jan/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

## Related Resources

- [GET Object \(p. 106\)](#)
- [PUT Object \(p. 129\)](#)
- [PUT Bucket \(p. 74\)](#)

# 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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### Response Elements

| Name                             | Description   |
|----------------------------------|---|
| <code>AccessControlList</code>   | Container for ACL information.<br>Type: Container<br>Ancestry: <code>AccessControlPolicy</code> |
| <code>AccessControlPolicy</code> | Container for the response.<br>Type: Container<br>Ancestry: None                                |

| Name        | Description   |
|-------------|---|
| 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 <i>ID</i> .<br>Type: String<br>Ancestry: AccessControlPolicy.Owner |
| Grant       | Container for <i>Grantee</i> and <i>Permission</i> .<br>Type: Container<br>Ancestry: AccessControlPolicy.AccessControlList  |
| Grantee     | Container for <i>DisplayName</i> and <i>ID</i> of the person being granted permissions.<br>Type: Container<br>Ancestry: AccessControlPolicy.AccessControlList.Grant   |
| ID          | Bucket owner's ID.<br>Type: String<br>Ancestry: AccessControlPolicy.Owner   |
| Owner       | Container for bucket owner information.<br>Type: Container<br>Ancestry: AccessControlPolicy   |
| Permission  | Permission given to the <i>Grantee</i> for bucket.<br>Type: String<br>Valid Values: FULL_CONTROL   WRITE   WRITE_ACP   READ   READ_ACP<br>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 \(p. 3\)](#).

## 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

### 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>8a6925ce4adee97f21c32aa379004fef</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>8a6925ce4adf57f21c32aa379004fef</ID>
        <DisplayName>CustomersName@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

## Related Resources

- [GET Bucket Objects \(p. 25\)](#)

# GET Bucket policy

## Description

This implementation of the `GET` operation uses the *policy* subresource to return the policy of a specified bucket. To use this operation, you must have `GetPolicy` permissions on the specified bucket, and you must be the bucket owner.

If you don't have `GetPolicy` permissions, Amazon S3 returns a 403 `Access Denied` error. If you have the correct permissions, but you're not the bucket owner, Amazon S3 returns a 405 `Method Not Allowed` error. If the bucket does not have a policy, Amazon S3 returns a 404 `Policy Not found` error. There are restrictions about who can create bucket policies and which objects in a bucket they can apply to. For more information, go to [Using Bucket Policies](#).

## Requests

### Syntax

```
GET /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### 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 \(p. 3\)](#).

## 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: AWS 02236Q3V0WHVSRSAMPLEX5sCYVf1bNRuU=
```

### 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": [ "1-22-333-4444", "3-55-678-9100" ]
      },
      "Action": [ "s3:*" ],
      "Resource": "arn:aws:s3:::bucket/*",
    }
  ]
}
```

## Related Resources

- [GET Bucket Objects \(p. 25\)](#)



# 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 \(p. 74\)](#).

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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### Response Elements

| Name                      | Description   |
|---------------------------|---|
| <i>LocationConstraint</i> | Specifies the Region where the bucket resides.<br>Type: String<br>Valid Values: EU   us-west-1   ap-southeast-1   ap-northeast-1   empty string (for the US Classic Region)<br>Ancestry: None |

When the bucket's Region is US Classic, 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 \(p. 3\)](#).

## 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: AWS 1ATXQ3HHA59CYF1CVS02:JUtd9kkJFjbKbkP9f6T/tAxozYY=
```

### Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint xmlns="http://s3.amazonaws.com/doc/2006-03-01/">EU</Loca
tionConstraint>
```

## Related Resources

- [GET Bucket Objects \(p. 25\)](#)
- [PUT Bucket \(p. 74\)](#)

# 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: signature
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### Response Elements

| Name                | Description  |
|---------------------|--|
| BucketLoggingStatus | Container for the response.<br>Type: Container<br>Ancestry: None |

| Name           | Description  |
|----------------|--|
| EmailAddress   | E-mail address of the person whose logging permissions are displayed.<br>Type: String<br>Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant.Grantee   |
| Grant          | Container for <i>Grantee</i> and <i>Permission</i> .<br>Type: Container<br>Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants   |
| Grantee        | Container for <i>EmailAddress</i> of the person whose logging permissions are displayed.<br>Type: Container<br>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.<br>Type: Container<br>Ancestry: BucketLoggingStatus |
| Permission     | Logging permissions assigned to the <i>Grantee</i> for the bucket.<br>Type: String<br>Valid Values: FULL_CONTROL   READ   WRITE<br>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.<br>Type: String<br>Ancestry: BucketLoggingStatus.LoggingEnabled     |
| TargetGrants   | Container for granting information.<br>Type: Container<br>Ancestry: BucketLoggingStatus.LoggingEnabled   |
| TargetPrefix   | Specifies the prefix for the keys that the log files are being stored under.<br>Type: String<br>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 \(p. 3\)](#).

## 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

### 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 \(p. 74\)](#)
- [PUT Bucket logging \(p. 84\)](#)

# GET Bucket notification

## Description

This implementation of the `GET` operation uses the *notification* subresource to return the notification configuration of a bucket. Currently, the *s3:ReducedRedundancyLostObject* event is the only event supported for notifications. The *s3:ReducedRedundancyLostObject* event is triggered when Amazon S3 detects that it has lost all replicas of a Reduced Redundancy Storage object and can no longer service requests for that object.

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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

## Response Elements

| Name                             | Description   |
|----------------------------------|---|
| <i>NotificationConfiguration</i> | Container for specifying the notification configuration of the bucket. If this element is empty, the bucket's notifications are turned off.<br>Type: Container<br>Children: <i>TopicConfiguration</i><br>Ancestry: None                               |
| <i>TopicConfiguration</i>        | Container for specifying the topic configuration for the notification. Currently, only one topic can be configured for notifications.<br>Type: Container<br>Children: <i>Topic</i> , <i>Event</i><br>Ancestry: <i>NotificationConfiguration</i>       |
| <i>Topic</i>                     | Amazon SNS topic to which Amazon S3 will publish a message to report the specified events for the bucket.<br>Type: String<br>Ancestry: <i>TopicConfiguration</i>  |
| <i>Event</i>                     | Bucket event to send notifications for. Currently, <i>s3:ReducedRedundancyLostObject</i> is the only event supported for notifications.<br>Type: String<br>Valid Values: <i>s3:ReducedRedundancyLostObject</i><br>Ancestry: <i>TopicConfiguration</i> |

## 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 \(p. 3\)](#).

## Examples

### Sample Request

This request returns the notification configuration on bucket quotes.s3.amazonaws.com.

```
GET ?notification HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 09 June 2010 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

### Sample Response

This response returns that the notification configuration for the specified bucket.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbiKa2bj0KMgUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A02
Date: Wed, 02 Jun 2010 12:00:00 GMT
Connection: close
Server: AmazonS3
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-east-1:123456789012:myTopic</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

## Related Resources

- [PUT Bucket notification \(p. 89\)](#)



# 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. 45).

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: signatureValue
```

### 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 |
|-------------------|--|----------|
| <i>delimiter</i>  | A delimiter is a character that you specify to group keys. All keys that contain the same string between the <i>prefix</i> and the first occurrence of the delimiter are grouped under a single result element in <i>CommonPrefixes</i> . These groups are counted as one result against the <i>max-keys</i> limitation. These keys are not returned elsewhere in the response. Also, see <i>prefix</i> .<br>Type: String<br>Default: None           | No       |
| <i>key-marker</i> | Specifies the key in the bucket that you want to start listing from. Also, see <i>version-id-marker</i> .<br>Type: String<br>Default: None   | No       |
| <i>max-keys</i>   | 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 <i>max-keys</i> was exceeded, the response contains <code>&lt;isTruncated&gt;true&lt;/isTruncated&gt;</code> . To return the additional keys, see <i>key-marker</i> and <i>version-id-marker</i> .<br>Type: String<br>Default: 1000 | No       |

| Parameter                | Description  | Required |
|--------------------------|--|----------|
| <i>prefix</i>            | 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 <i>prefix</i> to make groups in the same way you'd use a folder in a file system.) You can use <i>prefix</i> with <i>delimiter</i> to roll up numerous objects into a single result under <i>CommonPrefixes</i> . Also, see <i>delimiter</i> .<br>Type: String<br>Default: None | No       |
| <i>version-id-marker</i> | Specifies the object version you want to start listing from. Also, see <i>key-marker</i> .<br>Type: String<br>Default: None<br>Valid Values: Valid version ID   Default<br>Constraint: May not be an empty string  | No       |

## Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

## 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 \(p. 14\)](#).

### Response Elements

| Name         | Description   |
|--------------|---|
| DeleteMarker | Container for an object that is a Delete Marker.<br>Type: Container<br>Children: Key, VersionId, IsLatest, LastModified, Owner<br>Ancestor: ListVersionsResult                      |
| DisplayName  | Object owner's name.<br>Type: String<br>Ancestor: ListVersionsResult.Version.Owner  <br>ListVersionsResult.DeleteMarker.Owner   |
| ETag         | The entity tag is an MD5 hash of the object. The ETag only reflects changes to the contents of an object, not its metadata.<br>Type: String<br>Ancestor: ListVersionsResult.Version |

**Amazon Simple Storage Service API Reference**  
**GET Bucket Object versions**

---

| Name               | Description   |
|--------------------|---|
| ID                 | Object owner's ID.<br>Type: Boolean<br>Ancestor: ListVersionsResult.Version.Owner   ListVersionsResult.DeleteMarker.Owner   |
| IsLatest           | Specifies whether the object is ( <code>true</code> ) or is not ( <code>false</code> ) the latest version of an object.<br>Type: Boolean<br>Valid Values: <code>true</code>   <code>false</code><br>Ancestor: ListVersionsResult.Version   ListVersionsResult.DeleteMarker  |
| IsTruncated        | 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 <i>NextKeyMarker</i> and <i>NextVersionIdMarker</i> response parameters as a starting place in another request to return the rest of the results.<br>Type: Boolean<br>Valid Values: <code>true</code>   <code>false</code><br>Ancestor: ListVersionsResult |
| Key                | The object's key.<br>Type: String<br>Ancestor: ListVersionsResult.Version   ListVersionsResult.DeleteMarker   |
| KeyMarker          | Marks the last <i>Key</i> returned in a truncated response.<br>Type: String<br>Ancestor: ListVersionsResult   |
| LastModified       | Date and time the object was last modified.<br>Type: Date<br>Ancestor: ListVersionsResult.Version   ListVersionsResult.DeleteMarker   |
| ListVersionsResult | Container for the result.<br>Type: Container<br>Children: All elements in the response<br>Ancestor: ListVersionsResult  |
| MaxKeys            | Specifies the maximum number of objects to return.<br>Type: String<br>Default: 1000<br>Valid Values: Integers from 1 to 1000, inclusive<br>Ancestor: ListVersionsResult   |
| Name               | Bucket owner's name.<br>Type: String<br>Ancestor: ListVersionsResult  |

| Name                | Description  |
|---------------------|--|
| NextKeyMarker       | When the number of responses exceeds the value of <i>MaxKeys</i> , <i>NextKeyMarker</i> specifies the first key not returned that satisfies the search criteria. Use this value for the <i>key-marker</i> request parameter in a subsequent request.<br>Type: String<br>Ancestor: ListVersionsResult                         |
| NextVersionIdMarker | When the number of responses exceeds the value of <i>MaxKeys</i> , <i>NextVersionIdMarker</i> specifies the first object version not returned that satisfies the search criteria. Use this value for the <i>version-id-marker</i> request parameter in a subsequent request.<br>Type: String<br>Ancestor: ListVersionsResult |
| Owner               | Bucket owner.<br>Type: String<br>Children: DisplayName, ID<br>Ancestor: ListVersionsResult.Version   ListVersionsResult.DeleteMarker   |
| Prefix              | Selects objects that start with the value supplied by this parameter.<br>Type: String<br>Ancestor: ListVersionsResult  |
| Size                | Size in bytes of the object.<br>Type: String<br>Ancestor: ListVersionsResult.Version   |
| StorageClass        | Always STANDARD.<br>Type: String<br>Ancestor: ListVersionsResult.Version   |
| Version             | Container for version information.<br>Type: Container<br>Ancestor: ListVersionsResult  |
| VersionId           | Version ID of an object<br>Type: String<br>Ancestor: ListVersionsResult.Version   ListVersionsResult.DeleteMarker  |
| VersionIdMarker     | Marks the last version of the <i>Key</i> returned in a truncated response.<br>Type: String<br>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 \(p. 3\)](#).

## 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

### 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/L4kgtJl40Nr8X8gdRQBpUMLUo</VersionId>
    <IsLatest>>true</IsLatest>
    <LastModified>2009-10-12T17:50:30.000Z</LastModified>
    <ETag>"fba9dede5f27731c9771645a39863328"</ETag>
    <Size>434234</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>8a6925ce4adf58897f21c32aa379004fef</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Version>
  <DeleteMarker>
    <Key>my-second-image.jpg</Key>
    <VersionId>03jpff543dhffds434rfdSFdn943fdfsFkdmqnh892</VersionId>
    <IsLatest>>true</IsLatest>
    <LastModified>2009-11-12T17:50:30.000Z</LastModified>
    <Owner>
      <ID>8a6925ce4adf7f21c32aa379004fef</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>my-second-image.jpg</Key>
    <VersionId>QUpfndndhfd8438MNFdn93jdnJFkdmqnh893</VersionId>
    <IsLatest>>false</IsLatest>
    <LastModified>2009-10-10T17:50:30.000Z</LastModified>
    <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
    <Size>166434</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>8a6925ce421c32aa379004fef</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </Version>
</ListVersionsResult>
```

```
    </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>8a6925ce4adf532aa379004fef</ID>
      <DisplayName>mtd@amazon.com</DisplayName>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>my-third-image.jpg</Key>
    <VersionId>UIORUnfndfhnw89493jJFJ</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>8a6925ce4adf5887f21c32aa379004fef</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
User-Agent: curl/7.10.6 (i386-redhat-linux-gnu) libcurl/7.10.6 OpenSSL/0.9.7a
ipv6 zlib/1.1.4
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: AWS 0A1N5HNJ7ZX40NGOKT02:U1j5vNnJfzmiv3c1GnlG6MLVeZU=
```

## 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>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
<Size>217</Size>
<Owner>
  <ID>0fa3c2dd7e785a220ab556f8a401610b750c0b6d62ef87663dd1667a1e068364</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>0fa3c2dd7e785a2d1667a1e068364</ID>
  </Owner>
</DeleteMarker>
<Version>
  <Key>sourcekey</Key>
  <VersionId>wxXQ7ezLaL5JN2Sislq66Syxxo0k7uHTUpb9qiiMxNg.</VersionId>
  <IsLatest>false</IsLatest>
  <LastModified>2009-12-10T16:37:44.000Z</LastModified>
  <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;</ETag>
  <Size>217</Size>
  <Owner>
    <ID>0fa3c2dd7e785a220ab556f8a401610b750c0b6d62ef87663dd1667a1e068364</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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## 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>0fa3c2dd7e78dd1667a1e068364</ID>
</Owner>
</DeleteMarker>
<Version>
  <Key>sourcekey</Key>
  <VersionId>wxqQ7ezLaL5JN2Sislq66Syxxo0k7uHTUpb9qiiMxNg.</VersionId>
  <IsLatest>false</IsLatest>
  <LastModified>2009-12-10T16:37:44.000Z</LastModified>
  <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
  <Size>217</Size>
  <Owner>
    <ID>0fa3c2d87663dd1667a1e068364</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=t46ZenlYTBnj HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## 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>t46ZenlYTBnj</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>0fa3c2dd7e785a2201667a1e068364</ID>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxqQ7ezLaL5JN2Sislq66Syxxo0k7uHTUpb9qiiMxNg.</VersionId>
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z</LastModified>
    <ETag>"396fefef536d5ce46c7537ecf978a360"</ETag>
```



```
<Size>217</Size>
<Owner>
  <ID>0fa3c2dd7e785a663dd1667a1e068364</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*.

```
GET /?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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## 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>0fa3c2dd7e785a220ab667a1e068364</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>0fa3c2dd7e785a220ad1667a1e068364</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: AWS EEhpcyBtZXNzYWdlIHNPZ251ZCBieSB1bHZpbmc=
```

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>mvcbucketwithversionon1</Name>
  <Prefix></Prefix>
  <KeyMarker></KeyMarker>
  <VersionIdMarker></VersionIdMarker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>>false</IsTruncated>

  <Version>
    <Key>Sample.jpg</Key>
    <VersionId>toxMzQlBsGyGCz1YuMWmp90cdXLzqOCH</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: AWS VGhpcyBtZXNzYWdlIHNoZ25lZCBieSB1bHZpbmc=
```

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>3U275dAA4gz8ZOqOPhtJCU0i60krpCdy</VersionId>
    <IsLatest>true</IsLatest>
    <LastModified>2011-02-02T18:47:27.000Z</LastModified>
    <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
    <Size>0</Size>
    <Owner>
      <ID>852b113e7a2f25102679df27bb0ae12b3f85be6f290b936c4393484be31bebcc</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 \(p. 25\)](#)
- [GET Object \(p. 106\)](#)
- [PUT Object \(p. 129\)](#)
- [DELETE Object \(p. 102\)](#)

# 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: Signature
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

## 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 \(p. 14\)](#).

### Response Elements

| Name                               | Description   |
|------------------------------------|---|
| <i>Payer</i>                       | Specifies who pays for the download and request fees.<br>Type: Enum<br>Valid Values: Requester   BucketOwner<br>Ancestor: RequestPaymentConfiguration |
| <i>RequestPaymentConfiguration</i> | Container for <i>Payer</i> .<br>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 \(p. 3\)](#).

## 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
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 \(List Objects\) \(p. 25\)](#)

# 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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers](#) (p. 12).

### 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 \(p. 14\)](#).

### Response Elements

This implementation of GET returns the following response elements.

| Name                           | Description  |
|--------------------------------|--|
| <i>MfaDelete</i>               | Specifies whether MFA delete is enabled in the bucket versioning configuration. This element is only returned if the bucket has been configured with MfaDelete. If the bucket has never been so configured, this element is not returned.<br>Type: Enum<br>Valid Values: Disabled   Enabled<br>Ancestor: VersioningConfiguration |
| <i>Status</i>                  | The versioning state of the bucket.<br>Type: Enum<br>Valid Values: Suspended   Enabled<br>Ancestor: VersioningConfiguration  |
| <i>VersioningConfiguration</i> | Container for the <i>Status</i> response element.<br>Type: Container<br>Ancestor: None   |

### 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 \(p. 3\)](#).

## 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
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 \(p. 106\)](#)
- [PUT Object \(p. 129\)](#)
- [DELETE Object \(p. 102\)](#)

# GET Bucket website

## Description

This implementation of the GET operation returns the website configuration from the *website* subresource.

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.

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

## Requests

### Syntax

```
GET /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers](#) (p. 12).

### 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. 14).

### Response Elements

| Name                        | Description  |  |
|-----------------------------|--|--|
| <i>WebsiteConfiguration</i> | Container element<br>Type: Container<br>Ancestors: None  |  |
| <i>IndexDocument</i>        | Container for the <i>Suffix</i> element.<br>Type: Container<br>Ancestors: WebsiteConfiguration |  |

| Name                 | Description   |  |
|----------------------|---|--|
| <i>Suffix</i>        | 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)<br>Type: String<br>Ancestors: WebsiteConfiguration.IndexDocument |  |
| <i>ErrorDocument</i> | Container for <i>Key</i> elements<br>Type: Container<br>Ancestors: WebsiteConfiguration   |  |
| <i>Key</i>           | The object key name to use when a 4XX class error occurs<br>Type: String<br>Ancestors: WebsiteConfiguration.ErrorDocument   |  |

## 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 4DDSMRVYMBWZ1NWXWAG2: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 \(p. 23\)](#)
- [PUT Bucket website \(p. 99\)](#)

# 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, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon S3 Developer Guide*.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer Guide*.

## Requests

### Syntax

```
GET /?uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: Signature
```

### Request Parameters

| Parameter          | Description   | Required |
|--------------------|---|----------|
| <i>delimiter</i>   | Character you use to group keys.<br><br>All keys that contain the same string between the <i>prefix</i> , if specified, and the first occurrence of the delimiter after the prefix are grouped under a single result element, <i>CommonPrefixes</i> . If you don't specify the <i>prefix</i> parameter, then the substring starts at the beginning of the key. The keys that are grouped under <i>CommonPrefixes</i> result element are not returned elsewhere in the response.<br>Type: String | No       |
| <i>max-uploads</i> | 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.<br>Type: Integer<br>Default: 1,000  | No       |

| Parameter               | Description   | Required |
|-------------------------|---|----------|
| <i>key-marker</i>       | Together with <code>upload-id-marker</code> , this parameter specifies the multipart upload after which listing should begin.<br><br>If <code>upload-id-marker</code> is not specified, only the keys lexicographically greater than the specified <i>key-marker</i> will be included in the list.<br><br>If <code>upload-id-marker</code> is specified, any multipart uploads for a key equal to the <i>key-marker</i> might also be included, provided those multipart uploads have upload IDs lexicographically greater than the specified <i>upload-id-marker</i> .<br>Type: String | No       |
| <i>prefix</i>           | 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.)<br>Type: String  | No       |
| <i>upload-id-marker</i> | Together with <code>key-marker</code> , specifies the multipart upload after which listing should begin. If <code>key-marker</code> is not specified, the <code>upload-id-marker</code> parameter is ignored. Otherwise, any multipart uploads for a key equal to the <i>key-marker</i> might be included in the list only if they have an upload ID lexicographically greater than the specified <i>upload-id-marker</i> .<br>Type: String   | No       |

## Request Headers

This operation uses only request headers common to most requests. For more information, see [Common Request Headers \(p. 12\)](#).

## 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 \(p. 14\)](#).

### Response Elements

| Name                       | Description   |
|----------------------------|---|
| ListMultipartUploadsResult | Container for the response.<br>Children: <i>Bucket</i> , <i>KeyMarker</i> , <i>UploadIdMarker</i> , <i>NextKeyMarker</i> , <i>NextUploadIdMarker</i> , <i>MaxUploads</i> , <i>Delimiter</i> , <i>Prefix</i> , <i>CommonPrefixes</i> , <i>IsTruncated</i><br>Type: Container<br>Ancestor: None |

**Amazon Simple Storage Service API Reference**  
**List Multipart Uploads**

---

| Name               | Description   |
|--------------------|---|
| Bucket             | Name of the bucket to which the multipart upload was initiated.<br>Type: String<br>Ancestor: <i>ListMultipartUploadsResult</i>  |
| KeyMarker          | The key at or after which the listing began.<br>Type: String<br>Ancestor: <i>ListMultipartUploadsResult</i>   |
| UploadIdMarker     | Upload ID after which listing began.<br>Type: String<br>Ancestor: <i>ListMultipartUploadsResult</i>   |
| NextKeyMarker      | When a list is truncated, this element specifies the value that should be used for the <i>key-marker</i> request parameter in a subsequent request.<br>Type: String<br>Ancestor: <i>ListMultipartUploadsResult</i>  |
| NextUploadIdMarker | When a list is truncated, this element specifies the value that should be used for the <i>upload-id-marker</i> request parameter in a subsequent request.<br>Type: String<br>Ancestor: <i>ListMultipartUploadsResult</i>  |
| MaxUploads         | Maximum number of multipart uploads that could have been included in the response.<br>Type: Integer<br>Ancestor: <i>ListMultipartUploadsResult</i>  |
| IsTruncated        | Indicates whether the returned list of multipart uploads is truncated. A value of <i>true</i> indicates that the list was truncated. The list can be truncated if the number of multipart uploads exceeds the limit allowed or specified by <i>MaxUploads</i> .<br>Type: Boolean<br>Ancestor: <i>ListMultipartUploadsResult</i> |
| Upload             | Container for elements related to a particular multipart upload. A response can contain zero or more <i>Upload</i> elements.<br>Type: Container<br>Children: <i>Key</i> , <i>UploadId</i> , <i>InitiatorOwner</i> , <i>StorageClass</i> , <i>Initiated</i><br>Ancestor: <i>ListMultipartUploadsResult</i>                       |
| Key                | Key of the object for which the multipart upload was initiated.<br>Type: Integer<br>Ancestor: <i>Upload</i>   |
| UploadId           | Upload ID that identifies the multipart upload.<br>Type: Integer<br>Ancestor: <i>Upload</i>   |

**Amazon Simple Storage Service API Reference**  
**List Multipart Uploads**

---

| Name                              | Description   |
|-----------------------------------|---|
| 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 <i>Owner</i> element. If the initiator is an IAM User, then this element provides the user ARN and display name.</p> <p>Children: <i>ID, DisplayName</i></p> <p>Type: Container</p> <p>Ancestor: <i>Upload</i></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: <i>Initiator, Owner</i></p>   |
| DisplayName                       | <p>Principal's name.</p> <p>Type: String</p> <p>Ancestor: <i>Initiator, Owner</i></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 a the parent account ID and display name.</p> <p>Type: Container</p> <p>Children: <i>ID, DisplayName</i></p> <p>Ancestor: <i>Upload</i></p>  |
| StorageClass                      | <p>The class of storage (<code>STANDARD</code> or <code>REDUCED_REDUDANCY</code>) that will be used to store the object when the multipart upload is complete.</p> <p>Type: String</p> <p>Ancestor: <i>Upload</i></p>   |
| Initiated                         | <p>Date and time at which the multipart upload was initiated.</p> <p>Type: <i>Date</i></p> <p>Ancestor: <i>Upload</i></p>   |
| ListMultipartUploadsResult.Prefix | <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: <i>String</i></p> <p>Ancestor: <i>ListMultipartUploadsResult</i></p>  |
| 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: <i>String</i></p> <p>Ancestor: <i>ListMultipartUploadsResult</i></p>   |



| Name                               | Description   |
|------------------------------------|---|
| <code>CommonPrefixes</code>        | If you specify a delimiter in the request, then the result returns each distinct key prefix containing the delimiter in a <code>CommonPrefixes</code> element. The distinct key prefixes are returned in the <code>Prefix</code> child element.<br><br>Type: <i>Container</i><br>Ancestor: <i>ListMultipartUploadsResult</i>  |
| <code>CommonPrefixes.Prefix</code> | If the request does not include the <code>Prefix</code> 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.<br><br>If the request includes the <code>Prefix</code> parameter, then this element shows the substring of the key from the beginning to the first occurrence of the delimiter after the prefix.<br><br>Type: <i>String</i><br>Ancestor: <i>CommonPrefixes</i> |

## Special Errors

| Error Code                | Description  | HTTP Status Code | SOAP Fault Code Prefix |
|---------------------------|--|------------------|------------------------|
| <code>NoSuchUpload</code> | 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 \(p. 3\)](#).

## 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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZCBieSB1bHZpbmc=
```

### 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=YW55IGlkZWVgd2h5IGVsdmluZydzIHVwbG9hZCBmYW1sZWQ` (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: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/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>YW55IGlkZWVsdmluZydzIHVwbG9hZCBmYWlsZWQ</NextUp
loadIdMarker>
  <MaxUploads>3</MaxUploads>
  <IsTruncated>true</IsTruncated>
  <Upload>
    <Key>my-divisor</Key>
    <UploadId>XMgbGlrZSBlbHZpbmcncyBub3QgaGF2aW5nIG11Y2ggbHVjaw</UploadId>
    <Initiator>
      <ID>arn:aws:iam::111111111111:user/user1-11111a31-17b5-4fb7-9df5-
b11111f13de</ID>
      <DisplayName>user1-11111a31-17b5-4fb7-9df5-b11111f13de</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>VXBsb2FkIElEIGZvcjBlbHZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA</Up
loadId>
    <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>YW55IGlkZWVsdmluZydzIHVwbG9hZCBmYWlsZWQ</UploadId>
    <Initiator>
      <ID>arn:aws:iam::222222222222: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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZCBieSB1bHZpbmc=
```

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>Xgw4MJT6ZPAVx
pY0SAuGN7q4uWJm22ZYglW99trdp4tpO88.PT6.Mh00w2E17eutfAvQfQWoaJgE_W2gpcxQw--
</NextUploadIdMarker>
  <Delimiter></Delimiter>
  <Prefix/>
  <MaxUploads>1000</MaxUploads>
```

```
<IsTruncated>>false</IsTruncated>
<Upload>
  <Key>sample.jpg</Key>
  <UploadId>Agw4MJT6ZPAVxpY0SAuGN7q4uWJJM22ZYg1N99trdp4tp088.PT6.Mh00w2E17eut
fAvQfQWoaJgE_W2gpcxQw--</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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZCBieSB1bHZpbmc=
```

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 \(p. 149\)](#)
- [Upload Part \(p. 153\)](#)
- [Complete Multipart Upload \(p. 161\)](#)
- [Abort Multipart Upload \(p. 166\)](#)
- [List Parts \(p. 168\)](#)

# 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](#).

To configure the Region a bucket resides in, you use the `LocationConstraint` request element. 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 using `<CreateBucketConfiguration>`, applications that access your bucket must be able to handle 307 redirects.

To configure the access control settings for a bucket, use the `x-amz-acl` request header.

## Requests

### Syntax

```
PUT / HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: signatureValue
```

## 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. For more information, see [Common Request Headers \(p. 12\)](#).

| Name                   | Description   | Required |
|------------------------|---|----------|
| <code>x-amz-acl</code> | Sets the ACL of the bucket you're creating.<br>Type: String<br>Valid Values: private   public-read   public-read-write   authenticated-read   bucket-owner-read   bucket-owner-full-control<br>Default: private | No       |

## Request Elements

| Name                             | Description   | Required |
|----------------------------------|---|----------|
| <i>CreateBucketConfiguration</i> | Container for bucket configuration settings.<br>Type: Container<br>Ancestor: None   | No       |
| <i>LocationConstraint</i>        | Specifies the Region where the bucket will be created<br>Type: Enum<br>Valid Values: EU   us-west-1   ap-southeast-1   ap-northeast-1   empty string (for the US Classic Region)<br>Default: US Standard<br>Ancestor: CreateBucketConfiguration | No       |

## 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 \(p. 3\)](#).

## 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 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 124

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

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT

Location: /colourpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Sample Request Creating a Bucket and Setting the 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 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT

Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Object \(p. 129\)](#)



- [DELETE Bucket \(p. 19\)](#)

# PUT Bucket acl

## Description

This implementation of the `PUT` operation uses the `acl` subresource to set the access control list (ACL) permissions for an existing bucket. (To set the ACL permissions of a bucket when you create it, use the `x-amz-acl` request header.) To set the ACL of a bucket, you must have `WRITE_ACP` permission.

For more information about creating a bucket, see [PUT Bucket \(p. 74\)](#). For more information about returning the ACL of a bucket, see [GET Bucket ACL \(p. 112\)](#). For more information about setting the Versioning for a bucket, see [PUT Bucket Versioning Status \(p. 95\)](#).

## Requests

### Syntax

```
PUT /?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue

<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

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### Request Elements

| Name                     | Description  | Required |
|--------------------------|--|----------|
| <i>AccessControlList</i> | Container for Grant, Grantee, and Permission<br>Type: Container<br>Ancestors: AcessControlPolicy | No       |

| Name                       | Description  | Required |
|----------------------------|--|----------|
| <i>AccessControlPolicy</i> | Contains the elements that set the ACL permissions for an object per Grantee.<br>Type: String<br>Ancestors: None   | No       |
| <i>DisplayName</i>         | Screen name of the bucket owner.<br>Type: String<br>Ancestors: AccessControlPolicy.Owner   | No       |
| <i>Grant</i>               | Container for the grantee and his or her permissions.<br>Type: Container<br>Ancestors: AccessControlPolicy.AccessControlList   | No       |
| <i>Grantee</i>             | The subject whose permissions are being set. For more information, see <a href="#">Grantee Values (p. 79)</a> .<br>Type: String<br>Ancestors: AccessControlPolicy.AccessControlList.Grant    | No       |
| <i>ID</i>                  | ID of the bucket owner, or the ID of the grantee.<br>Type: String<br>Ancestors: AccessControlPolicy.Owner   AccessControlPolicy.AccessControlList.Grant                                      | No       |
| <i>Owner</i>               | Container for the bucket owner's display name and ID.<br>Type: Container<br>Ancestors: AccessControlPolicy   | No       |
| <i>Permission</i>          | Specifies the permission given to the grantee.<br>Type: String<br>Valid Values: FULL_CONTROL   WRITE   WRITE_ACP   READ   READ_ACP<br>Ancestors: AccessControlPolicy.AccessControlList.Grant | No       |

### Grantee Values

You can specify the person (grantee) that you're assigning access rights to (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>ID</ID>
  <DisplayName>GranteesEmail</DisplayName>
</Grantee>
```

*DisplayName* is optional and ignored in the request.

- By E-mail address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AmazonCustomerByEmail">
  <EmailAddress>Grantees@email.com</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>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</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 \(p. 14\)](#).

### 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 \(p. 3\)](#).

## Examples

### Sample Request

This request gives the owner of the bucket full control.

```
PUT ?acl HTTP/1.1
Host: quotes.s3.amazonaws.com
Content-Length: 214
Date: Wed, 25 Nov 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=

<AccessControlPolicy>
  <Owner>
    <ID>8a6925ce4adf588a4f21c32aa379004fef</ID>
    <DisplayName>BucketOwnersEmail@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>8a6925ce47f21c32aa379004fef</ID>
```

```
<DisplayName>BucketOwnersEmail@amazon.com</DisplayName>
</Grantee>
<Permission>FULL_CONTROL</Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFBiKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
```

## Related Resources

- [PUT Bucket \(p. 74\)](#)
- [DELETE Bucket \(p. 19\)](#)

# PUT Bucket policy

## Description

This implementation of the `PUT` operation uses the *policy* subresource to add to or replace a policy on a bucket. If the bucket already has a policy, the one in this request completely replaces it. To perform this operation, you must be the bucket owner or someone authorized by the bucket owner to set a policy on the bucket, and have *PutPolicy* permissions.

If you don't have *PutPolicy* permissions on the bucket, Amazon S3 returns a *403 AccessDenied* error. There are restrictions about who can create bucket policies and which objects in a bucket they can apply to. For more information, go to [Using Bucket Policies](#).

## Requests

### Syntax

```
PUT /?policy HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue

Policy written in JSON
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers](#) (p. 12).

### 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. 14).

### 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. 3).

## 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: AWS VGhpcyBtdlISAMPLEBlbHZpbmc=

{
  "Version": "2008-10-17",
  "Id": "aaaa-bbbb-cccc-dddd",
  "Statement" : [
    {
      "Effect": "Allow",
      "Sid": "1",
      "Principal" : {
        "AWS": [ "1-22-333-4444", "3-55-678-9100" ]
      },
      "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 \(p. 74\)](#)
- [DELETE Bucket \(p. 19\)](#)

# PUT Bucket logging

## Description



### Note

The logging implementation of PUT Bucket is a beta feature.

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. 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 creating a bucket, see [PUT Bucket \(p. 74\)](#). For more information about returning the logging status of a bucket, see [GET Bucket logging \(p. 39\)](#).

## Requests

### Syntax

```
PUT /?logging HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

*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 only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### Request Elements

| Name                       | Description   | Required |
|----------------------------|---|----------|
| <i>BucketLoggingStatus</i> | Container for logging status information.<br>Type: Container<br>Children: <i>LoggingEnabled</i><br>Ancestry: None | Yes      |



**Amazon Simple Storage Service API Reference**  
**PUT Bucket logging**

---

| Name                  | Description  | Required |
|-----------------------|--|----------|
| <i>EmailAddress</i>   | E-mail address of the person being granted logging permissions.<br>Type: String<br>Children: None<br>Ancestry:<br>BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant.Grantee  | No       |
| <i>Grant</i>          | Container for the grantee and his/her logging permissions.<br>Type: Container<br>Children: Grantee, Permission<br>Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants  | No       |
| <i>Grantee</i>        | Container for <i>EmailAddress</i> of the person being granted logging permissions. For more information, see <a href="#">Grantee Values (p. 86)</a> .<br>Type: Container<br>Children: EmailAddress<br>Ancestry:<br>BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant   | No       |
| <i>LoggingEnabled</i> | Container for logging information. This element is present when you are enabling logging (and not present when you are disabling logging).<br>Type: Container<br>Children: Grant, TargetBucket, TargetPrefix<br>Ancestry: BucketLoggingStatus  | No       |
| <i>Permission</i>     | Logging permissions given to the <i>Grantee</i> for the bucket. The bucket owner is automatically granted FULL_CONTROL to all logs delivered to the bucket. This optional element enables you grant access to others.<br>Type: String<br>Valid Values: FULL_CONTROL   READ   WRITE<br>Children: None<br>Ancestry:<br>BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant   | No       |
| <i>TargetBucket</i>   | Specifies the bucket where you want Amazon S3 to store server access logs. You can have your logs delivered to any bucket that you own, including the same bucket that is being logged. 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.<br>Type: String<br>Children: None<br>Ancestry: BucketLoggingStatus.LoggingEnabled | No       |

| Name                | Description  | Required |
|---------------------|--|----------|
| <i>TargetGrants</i> | Container for granting information.<br>Type: Container<br>Children: Grant, Permission<br>Ancestry: BucketLoggingStatus.LoggingEnabled  | No       |
| <i>TargetPrefix</i> | This element lets you specify a prefix for the keys that the log files will be stored under.<br>Type: String<br>Children: None<br>Ancestry: BucketLoggingStatus.LoggingEnabled | No       |

## Grantee Values

You can specify the person (grantee) that you're assigning access rights to (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>ID</ID>
  <DisplayName>GranteesEmail</DisplayName>
</Grantee>
```

*DisplayName* is optional and ignored in the request.

- By E-mail address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AmazonCustomerByEmail">
  <EmailAddress>Grantees@email.com</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>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</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 \(p. 14\)](#).

## 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 \(p. 3\)](#).

## 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=

<?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/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

```
<?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/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
```

## Related Resources

- [PUT Object \(p. 129\)](#)
- [DELETE Bucket \(p. 19\)](#)
- [PUT Bucket \(p. 74\)](#)
- [GET Bucket logging \(p. 39\)](#)

# PUT Bucket notification

## Description

This implementation of the `PUT` operation uses the *notification* subresource to enable notifications of specified events for a bucket. Currently, the *s3:ReducedRedundancyLostObject* event is the only event supported for notifications. The *s3:ReducedRedundancyLostObject* event is triggered when Amazon S3 detects that it has lost all replicas of an object and can no longer service requests for that object.

If the bucket owner and Amazon SNS topic owner are the same, the bucket owner has permission to publish notifications to the topic by default. Otherwise, the owner of the topic must create a policy to enable the bucket owner to publish to the topic. For more information about creating this policy, go to [Example Cases for Amazon SNS Access Control](#).

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.

After you call the `PUT` operation to configure notifications on a bucket, Amazon S3 publishes a test notification to ensure that the topic exists and that the bucket owner has permission to publish to the specified topic. If the notification is successfully published to the SNS topic, the `PUT` operation updates the bucket configuration and returns the 200 OK response with a *x-amz-sns-test-message-id* header containing the message ID of the test notification sent to topic.

To turn off notifications on a bucket, you specify an empty *NotificationConfiguration* element in your request: `<NotificationConfiguration />`

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

```
PUT /?notification HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>TopicARN</Topic>
    <Event>Event</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers](#) (p. 12).

## Request Elements

| Name                             | Description  | Required |
|----------------------------------|--|----------|
| <i>NotificationConfiguration</i> | Container for specifying the notification configuration of the bucket. If this element is empty, notifications are turned off on the bucket.<br>Type: Container<br>Children: <i>TopicConfiguration</i><br>Ancestry: None                                       | Yes      |
| <i>TopicConfiguration</i>        | Container for specifying the topic configuration for the notification. Currently, only one topic can be configured for notifications.<br>Type: Container<br>Children: <i>Topic</i> , <i>Event</i><br>Ancestry: <i>NotificationConfiguration</i>                | No       |
| <i>Topic</i>                     | Amazon SNS topic to which Amazon S3 will publish a message to report the specified events for the bucket.<br>Type: String<br>Ancestry: <i>TopicConfiguration</i>   | No       |
| <i>Event</i>                     | Bucket event for which to send notifications. Currently, <i>s3:ReducedRedundancyLostObject</i> is the only event supported for notifications.<br>Type: String<br>Valid Values:<br><i>s3:ReducedRedundancyLostObject</i><br>Ancestry: <i>TopicConfiguration</i> | 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 \(p. 14\)](#).

### 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 | InvalidArgument | <p>The following conditions can cause this error:</p> <ul style="list-style-type: none"> <li>• The specified event is not supported for notifications.</li> <li>• The specified topic ARN does not exist or is not well-formed. Verify the topic ARN.</li> <li>• The specified topic is in a different region than the bucket. You must use a topic that resides in the same Region as the bucket.</li> <li>• The bucket owner does not have <i>Publish</i> permission on the specified topic.</li> </ul> |
| HTTP 403 Forbidden   | AccessDenied    | <p>You are not the owner of the specified bucket or you do not have the <i>s3:PutBucketNotification</i> bucket permission to set the notification configuration on the bucket.</p>  |

For general information about Amazon S3 errors and a list of error codes, see [Error Responses \(p. 3\)](#).

## Examples

### Sample Requests

This request enables notification on bucket `quotes.s3.amazonaws.com` for the event *s3:ReducedRedundancyLostObject* with notifications published to the topic `arn:aws:sns:us-east-1:123456789012:myTopic`.

```
PUT ?notification HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 02 Jun 2010 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
<NotificationConfiguration>
  <TopicConfiguration>
    <Topic>arn:aws:sns:us-east-1:123456789012:myTopic</Topic>
    <Event>s3:ReducedRedundancyLostObject</Event>
  </TopicConfiguration>
</NotificationConfiguration>
```

This request turns off notification on the `quotes.s3.amazonaws.com` bucket.

```
PUT ?notification HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 02 Jun 2010 12:01:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
<NotificationConfiguration />
```

### Sample Responses

In this response, you are notified that the notification configuration was successful. It also returns the ID of the test message Amazon S3 sent to the topic.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbIKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
x-amz-sns-test-message-id: feebldff-cc96-449d-964c-f8a1890fd007
Date: Wed, 02 Jun 2010 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

This response returns that the notification was turned off successfully. Note that Amazon S3 doesn't send a test notification when notifications are turned off.

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPiFbIKa2bj0KMgUAdQkf3ShJTOOpXUueF6QKo
x-amz-request-id: 236A890524860101
Date: Wed, 02 Jun 2010 12:01:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [GET Bucket notification \(p. 42\)](#)



# 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 only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### Request Elements

| Name                               | Description   |
|------------------------------------|---|
| <i>Payer</i>                       | Specifies who pays for the download and request fees.<br>Type: Enum<br>Valid Values: Requester   BucketOwner<br>Ancestor: RequestPaymentConfiguration |
| <i>RequestPaymentConfiguration</i> | Container for <i>Payer</i> .<br>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 \(p. 14\)](#).

## 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 \(p. 3\)](#).

## 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 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=

<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 2009 12:00:00 GMT
Location: /colorpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Bucket \(p. 74\)](#)
- [GET Bucket requestPayment \(p. 57\)](#)

# 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.

For more information about creating a bucket, see [PUT Bucket \(p. 74\)](#). For more information about returning the versioning state of a bucket, see [GET Bucket Versioning Status \(p. 59\)](#).

## Requests

### Syntax

```
PUT /?versioning HTTP/1.1
Host: BucketName.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: signatureValue
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    |
|------------------|---|-------------|
| <i>x-amz-mfa</i> | The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device.<br>Type: String<br>Default: None<br>Condition: Required to configure the versioning state if versioning is configured with MFA Delete enabled. | Conditional |

## Request Elements

| Name                           | Description  | Required |
|--------------------------------|--|----------|
| <i>Status</i>                  | Sets the versioning state of the bucket.<br>Type: Enum<br>Valid Values: Suspended   Enabled<br>Ancestor: VersioningConfiguration   | No       |
| <i>MfaDelete</i>               | Specifies whether MFA Delete is enabled in the bucket versioning configuration. When enabled, the bucket owner must include the <i>x-amz-mfa</i> request header in requests to change the versioning state of a bucket and to permanently delete a versioned object.<br>Type: Enum<br>Valid Values: Disabled   Enabled<br>Ancestor: VersioningConfiguration<br>Constraint: Can only be used when you use <i>Status</i> . | No       |
| <i>VersioningConfiguration</i> | Container for setting the versioning state.<br>Type: Container<br>Children: Status<br>Ancestor: None   | 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. 14).

### 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 \(p. 3\)](#).

## Examples

### Sample Request

The following request enables 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
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/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
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/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
```

## Sample Request Enabling Versioning and MFA Delete on a Bucket

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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
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/0zo3emzU4dzsD4rcKCHQUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT

Location: /colourpictures
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [DELETE Bucket \(p. 19\)](#)
- [PUT Bucket \(p. 74\)](#)

# PUT Bucket website

## Description

This implementation of the PUT operation uses the *website* subresource to set the website configuration.

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 granting them the `S3:PutBucketWebsite` permission.

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

## Requests

### Syntax

```
PUT /?website HTTP/1.1
Host: bucketname.s3.amazonaws.com
Date: date
Content-Length: ContentLength
Authorization: signatureValue

<WebsiteConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>errorDocument.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### Request Elements

| Name                        | Description  | Required |
|-----------------------------|--|----------|
| <i>WebsiteConfiguration</i> | Container for the request<br>Type: Container<br>Ancestors: None                                | Yes      |
| <i>IndexDocument</i>        | Container for the <i>Suffix</i> element.<br>Type: Container<br>Ancestors: WebsiteConfiguration | Yes      |

| Name                 | Description   | Required    |
|----------------------|---|-------------|
| <i>Suffix</i>        | 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 <code>index.html</code> and you make a request to <code>samplebucket/images/</code> the data that is returned will be for the object with the key name <code>images/index.html</code> )<br>The suffix must not be empty and must not include a slash character.<br>Type: String<br>Ancestors: WebsiteConfiguration.IndexDocument | Yes         |
| <i>ErrorDocument</i> | Container for Key element<br>Type: Container<br>Ancestors: WebsiteConfiguration   | No          |
| <i>Key</i>           | The object key name to use when a 4XX class error occurs<br>Type: String<br>Ancestors: WebsiteConfiguration.ErrorDocument<br>Condition: Required when <i>ErrorDocument</i> is specified.  | Conditional |

## 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 \(p. 14\)](#).

### Response Elements

This implementation of the operation does not return response elements.

## Examples

### Sample Request

This request configures a bucket as a website by adding the website subresource to the bucket. The website configuration specifies the index document name, `index.html`, and error document, `404.html`.

```
PUT ?website HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Content-Length: 242
Date: Thu, 27 Jan 2011 12:00:00 GMT
Authorization: AWS 1DBSMRRRRRWZ1NWYWAG2:acxI7sWO+ugzxhf2AtcqRLGy70B=

<WebsiteConfiguration xmlns='http://s3.amazonaws.com/doc/2006-03-01/'>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <ErrorDocument>
    <Key>404.html</Key>
  </ErrorDocument>
</WebsiteConfiguration>
```



## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJT0OpXUueF6QKo
x-amz-request-id: 80CD4368BD211111
Date: Thu, 27 Jan 2011 00:00:00 GMT
Content-Length: 0
Server: AmazonS3
```

## Related Resources

- [DELETE Bucket website](#) (p. 23)
- [GET Bucket website](#) (p. 62)

# Operations on Objects

### Topics

- [DELETE Object](#) (p. 102)
- [GET Object](#) (p. 106)
- [GET Object ACL](#) (p. 112)
- [GET Object torrent](#) (p. 116)
- [HEAD Object](#) (p. 118)
- [POST Object](#) (p. 122)
- [PUT Object](#) (p. 129)
- [PUT Object acl](#) (p. 135)
- [PUT Object - Copy](#) (p. 142)
- [Initiate Multipart Upload](#) (p. 149)
- [Upload Part](#) (p. 153)
- [Upload Part - Copy](#) (p. 156)
- [Complete Multipart Upload](#) (p. 161)
- [Abort Multipart Upload](#) (p. 166)
- [List Parts](#) (p. 168)

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



### Note

For information about access policies, see [REST Access Policy](#) .

# 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 latest 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 \(p. 104\)](#).

## Requests

### Syntax

```
DELETE /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Content-Length: length
Authorization: signatureValue
```

## Request Parameters

This implementation of the operation does not use request parameters.

## Request Headers

| Name                   | Description  | Required    |
|------------------------|--|-------------|
| <code>x-amz-mfa</code> | The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device.<br>Type: String<br>Default: None<br>Condition: Required to permanently delete a versioned object if versioning is configured with MFA Delete enabled. | Conditional |

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

| Header              | Description   |
|---------------------|---|
| x-amz-delete-marker | Specifies whether the versioned object that was permanently deleted was ( <code>true</code> ) or was not ( <code>false</code> ) a delete marker. In a simple DELETE, this header indicates whether ( <code>true</code> ) or not ( <code>false</code> ) a delete marker was created.<br>Type: Boolean<br>Valid Values: <code>true</code>   <code>false</code><br>Default: <code>false</code> |
| x-amz-version-id    | 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.<br><br>Type: String<br><br>Default: None   |

### 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 \(p. 3\)](#).

## 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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
```

### Sample Response

```
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
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=UIORUnfndfiufdisojhr398493jfdkjFJjkndnqUif
hnw89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

## Sample Response

```
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: UIORUnfndfiufdisojhr398493jfdkjFJjkndnqUifhnw89493jJFJ
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: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 3/L4kqtJlcpXroDTdMJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUM
LUo
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: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

## Sample Response

```
HTTPS/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEbla0wiIOIxl+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 \(p. 129\)](#)
- [DELETE Object \(p. 102\)](#)

# 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.

To distribute large files to many people, you can save bandwidth costs 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. 112).

## Versioning

By default, the `GET` operation returns the latest version of an object. To return a different version, use the `versionId` subresource.



### Note

If the latest 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. 95). To see sample requests that use versioning, see [Sample Request Getting a Specified Version of an Object](#) (p. 109).

## Requests

### Syntax

```
GET /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
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 only sent 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 Pre-signed URL, when using these parameters. They can not be used with an unsigned (anonymous) request.

| Parameter                           | Description   | Required |
|-------------------------------------|---|----------|
| <i>response-content-type</i>        | Sets the Content-Type header of the response.<br>Type: String<br>Default: None        | No       |
| <i>response-content-language</i>    | Sets the Content-Language header of the response.<br>Type: String<br>Default: None    | No       |
| <i>response-expires</i>             | Sets the Expires header of the response.<br>Type: String<br>Default: None             | No       |
| <i>response-cache-control</i>       | Sets the Cache-Control header of the response.<br>Type: String<br>Default: None       | No       |
| <i>response-content-disposition</i> | Sets the Content-Disposition header of the response.<br>Type: String<br>Default: None | No       |
| <i>response-content-encoding</i>    | Sets the Content-Encoding header of the response.<br>Type: String<br>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. For more information, see [Common Request Headers \(p. 12\)](#).

| Name                       | Description   | Required |
|----------------------------|---|----------|
| <i>Range</i>               | 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> .<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>If-Modified-Since</i>   | Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).<br>Type: String<br>Default: None<br>Constraints: None   | No       |
| <i>If-Unmodified-Since</i> | Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).<br>Type: String<br>Default: None<br>Constraints: None  | No       |

| Name                 | Description  | Required |
|----------------------|--|----------|
| <i>If-Match</i>      | Return the object only if its entity tag ( <i>ETag</i> ) is the same as the one specified, otherwise return a 412 (precondition failed).<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>If-None-Match</i> | Return the object only if its entity tag ( <i>ETag</i> ) is different from the one specified, otherwise return a 304 (not modified).<br>Type: String<br>Default: None<br>Constraints: None     | No       |

## Request Elements

This implementation of the operation does not use request elements.

## Responses

### Response Headers

| Header                              | Description   |
|-------------------------------------|---|
| x-amz-delete-marker                 | Specifies whether the object retrieved was ( <i>true</i> ) or was not ( <i>false</i> ) a Delete Marker. If false, this response header does not appear in the response.<br>Type: Boolean<br>Valid Values: <i>true</i>   <i>false</i><br>Default: <i>false</i> |
| <i>x-amz-server-side-encryption</i> | If the object is stored using server-side encryption, response includes this header with value of the encryption algorithm used.<br>Type: String<br>Valid Values: <i>AES256</i>   |
| x-amz-version-id                    | Returns the version ID of the retrieved object if it has a unique version ID.<br>Type: String<br>Default: None  |

## 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 \(p. 3\)](#).



## 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: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9ASled4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
[434234 bytes of object data]
```

### Sample Response if 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

### 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 KSIAJ7YG7FXPX2U2PYAA: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+Il3/QqiulZKEuegzLAAspwsqwnwygb9GgFseeFHL5CII8NXSrfWW2
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: "0332beela7bf845f176c5c0d1ae7cf07"
Accept-Ranges: bytes
Content-Type: text/plain
Content-Length: 22
Server: AmazonS3

[object data not shown]
```

## Sample Request with the Range Header

The following request specifies the HTTP Range header to retrieve first 10 bytes of an object. For more information about the HTTP Range header, go to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.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 EEIAJ7YG7FXPX2U2PYAA:Yxg83MzaEgh3OZ3l0rLo5RTX1lo=
Sample Response with Specified Range of the Object Bytes
```

In the following sample response note the header values are set to the values specified in the `true` request.

```
HTTP/1.1 206 Partial Content
x-amz-id-2: MzRISOwyjmnupCzjI1WC06l5TTAzm7/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]
```

## Related Resources

- [GET Service \(p. 15\)](#)
- [GET Object acl \(p. 112\)](#)

# 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 latest 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. 114).

## Requests

### Syntax

```
GET /ObjectName?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
Range:bytes=byte_range
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers](#) (p. 12).

### 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. 14).

### Response Elements

| Name                           | Description   |
|--------------------------------|---|
| <code>AccessControlList</code> | Container for Grant, Grantee, and Permission<br>Type: Container<br>Ancestors: AccessControlPolicy |

| Name                | Description  |
|---------------------|--|
| AccessControlPolicy | Contains the elements that set the ACL permissions for an object per Grantee.<br>Type: Container<br>Ancestors: None  |
| DisplayName         | Screen name of the bucket owner<br>Type: String<br>Ancestors: AccessControlPolicy.Owner  |
| Grant               | Container for the grantee and his or her permissions.<br>Type: Container<br>Ancestors: AccessControlPolicy.AccessControlList                               |
| Grantee             | The subject whose permissions are being set.<br>Type: String<br>Ancestors: AccessControlPolicy.AccessControlList.Grant                                     |
| ID                  | ID of the bucket owner, or the ID of the grantee<br>Type: String<br>Ancestors: AccessControlPolicy.Owner or<br>AccessControlPolicy.AccessControlList.Grant |
| Owner               | Container for the bucket owner's display name and ID.<br>Type: Container<br>Ancestors: AccessControlPolicy   |
| Permission          | Specifies the permission (FULL_CONTROL, WRITE, READ_ACP) given to the grantee.<br>Type: String<br>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 \(p. 3\)](#).

## 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 4HL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40NrjfkD
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>8a6925ce4adf588a4532aa379004fef</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>8a6925ce4adf588a453214a379004fef</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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## 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+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3

<AccessControlPolicy>
```

```
<Owner>
  <ID>8a6925ce4adf588a421c32aa379004fef</ID>
  <DisplayName>mdtd@amazon.com</DisplayName>
</Owner>
<AccessControlList>
  <Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
      <ID>8a6925ce4adf588a4532142d3f74dd8c71fa124b1dde97f21c32aa379004fef</ID>

      <DisplayName>mdtd@amazon.com</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>
```

## Related Resources

- [GET Object \(p. 106\)](#)
- [PUT Object \(p. 129\)](#)
- [DELETE Object \(p. 102\)](#)

# 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.

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: signatureValue
```

### Request Parameters

This implementation of the operation does not use request parameters.

### Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see [Common Request Headers \(p. 12\)](#).

### 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 \(p. 14\)](#).

### 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 \(p. 3\)](#).



## 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

### 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 \(p. 106\)](#)

# HEAD Object

## Description

The `HEAD` operation retrieves metadata from an object without returning the object itself. This operation is useful if you're only interested 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.

## Versioning

By default, the `HEAD` operation retrieves metadata from the latest version of an object. (If the latest 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. 120).

## Requests

### Syntax

```
HEAD /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Authorization: signatureValue
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. For more information, see [Common Request Headers](#) (p. 12).

| Name                     | Description   | Required |
|--------------------------|---|----------|
| <i>Range</i>             | 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> .<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>If-Modified-Since</i> | Return the object only if it has been modified since the specified time, otherwise return a 304 (not modified).<br>Type: String<br>Default: None<br>Constraints: None   | No       |

| Name                       | Description  | Required |
|----------------------------|--|----------|
| <i>If-Unmodified-Since</i> | Return the object only if it has not been modified since the specified time, otherwise return a 412 (precondition failed).<br>Type: String<br>Default: None<br>Constraints: None               | No       |
| <i>If-Match</i>            | Return the object only if its entity tag ( <i>ETag</i> ) is the same as the one specified, otherwise return a 412 (precondition failed).<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>If-None-Match</i>       | Return the object only if its entity tag ( <i>ETag</i> ) is different from the one specified, otherwise return a 304 (not modified).<br>Type: String<br>Default: None<br>Constraints: None     | 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 \(p. 14\)](#).

| Name                                | Description  |  |
|-------------------------------------|--|--|
| <i>x-amz-meta-*</i>                 | If you supplied user metadata when you <code>PUT</code> the object, that metadata is returned in one or more response headers prefixed with <code>x-amz-meta-</code> and with the suffix name that you provided on storage, for example, <code>family</code> , i.e., <code>x-amz-meta-family</code> . Amazon S3 returns this metadata verbatim; Amazon S3 does not interpret it.<br>Type: String |  |
| <i>x-amz-missing-meta</i>           | This is set to the number of metadata entries not returned in <code>x-amz-meta</code> headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers.<br>Type: String   |  |
| <i>x-amz-server-side-encryption</i> | If the object is stored using server-side encryption, response includes this header with value of the encryption algorithm used.<br>Type: String<br>Valid Values: <code>AES256</code>  |  |

| Name                    | Description   |  |
|-------------------------|---|--|
| <i>x-amz-version-id</i> | This is set to the version ID of the object returned.<br>Type: String |  |

## 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 \(p. 3\)](#).

## Examples

### Sample Request

This 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 02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

### Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXroDTDmjVBH40NrjfkD
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 Getting Metadata From a Specified Version of an Object

This operation returns the metadata of the specified version of an object.

```
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40NrjfkD HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS 02236Q3V0WpaBX5sCYVf1bNRuU=
```

## Sample Response to a Versioned HEAD Request

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8epIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXrof3vjVBH40NrjfkD
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
```

## Related Resources

- [GET Object \(p. 106\)](#)

# 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 will be overwritten.

To ensure that data is not corrupted traversing the network, use the Content-MD5 form field. When you use the Content-MD5 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 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 prior to 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 (e.g., 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>.

## 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. 59).

Amazon S3 is a distributed system. If you enable versioning on a bucket and Amazon S3 receives multiple write requests for the same object simultaneously, all of the objects will be stored.

To see sample requests that use versioning, see [Sample Request](#) (p. 127).

## Requests

### Syntax

```
POST /KeyName 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="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.

**Amazon Simple Storage Service API Reference**  
**POST Object**

---

| Name  | Description  | Required    |
|---|--|-------------|
| <i>AWSSecretKeyId</i>   | 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.<br>Type: String<br>Default: None<br>Constraints: Required if a policy document is included with the request.   | Conditional |
| <i>Cache-Control, Content-Type, Content-Disposition, Content-Encoding</i> | REST-specific headers.<br>For more information, see <a href="#">PUT Object (p. 129)</a> .<br>Type: String<br>Default: None   | No          |
| <i>expires</i>  | Number of milliseconds before expiration<br>Type: Int<br>Default: None   | No          |
| <i>file</i>   | File or text content.<br>The file or text content must be the last field in the form.<br>You cannot upload more than one file at a time.<br>Type: File or text content<br>Default: None  | Yes         |
| <i>key</i>  | The name of the uploaded key.<br>To use the filename provided by the user, use the <code>\${filename}</code> variable. For example, if the user Betty uploads the file lolcatz.jpg and you specify <code>/user/betty/\${filename}</code> , the key name will be <code>/user/betty/lolcatz.jpg</code> .<br>For more information, go to <a href="#">Using Keys</a> in the <a href="#">Amazon Simple Storage Service Developer Guide</a> .<br>Type: String<br>Default: None | Yes         |
| <i>policy</i>   | Security Policy describing what is permitted in the request. Requests without a security policy are considered anonymous and only work on publicly writable buckets.<br>Type: String<br>Default: None  | No          |



| Name                                     | Description  | Required |
|--|--|----------|
| <i>success_action_redirect, redirect</i> | <p>The URL to which the client is redirected upon successful upload.</p> <p>If <i>success_action_redirect</i> is not specified, Amazon S3 returns the empty document type specified in the <i>success_action_status</i> 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<br/>Default: None</p> <div>  <b>Note</b> <p>The <i>redirect</i> field name is deprecated and support for the <i>redirect</i> field name will be removed in the future.</p> </div>   | No       |
| <i>success_action_status</i>             | <p>The status code returned to the client upon successful upload if <i>success_action_redirect</i> is not specified.</p> <p>Accepts the values 200, 201, or 204 (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<br/>Default: None</p> <div>  <b>Note</b> <p>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 <i>success_action_status</i> to 201.</p> </div> | No       |
| <i>x-amz-storage-class</i>               | <p>Storage class to use for storing the object.</p> <p>Type: String<br/>Default: STANDARD<br/>Valid Values: STANDARD   REDUCED_REDUNDANCY</p>  | No       |

| Name                                | Description  | Required |
|-------------------------------------|--|----------|
| <i>x-amz-meta-*</i>                 | Field names prefixed with <i>x-amz-meta-</i> contain user-specified metadata.<br>Amazon S3 does not validate or use this data.<br>For more information, see <a href="#">PUT Object (p. 129)</a> .<br>Type: String<br>Default: None   | No       |
| <i>x-amz-security-token</i>         | Amazon DevPay security token.<br>Each request that uses Amazon DevPay requires two <i>x-amz-security-token</i> form fields: one for the product token and one for the user token.<br>For more information, go to <a href="#">Using DevPay</a> .<br>Type: String<br>Default: None | No       |
| <i>x-amz-server-side-encryption</i> | Specifies server-side encryption algorithm to use when Amazon S3 creates an object.<br>Type: String<br>Valid Value: AES256   | No       |

## 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 \(p. 14\)](#).

| Name                                     | Description  |
|--|--|
| <i>success_action_redirect, redirect</i> | The URL to which the client is redirected on successful upload.<br>Type: String<br>Ancestor: PostResponse  |
| <i>x-amz-server-side-encryption</i>      | If you request server-side encryption when adding an object, the response includes this header confirming the encryption algorithm used.<br>Type: String |
| <i>x-amz-version-id</i>                  | Version of the object.<br>Type: String   |

### Response Elements

| Name          | Description  |
|---------------|--|
| <i>Bucket</i> | Name of the bucket the object was stored in.<br>Type: String<br>Ancestor: PostResponse |

| Name            | Description  |
|-----------------|--|
| <i>ETag</i>     | 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. The ETag only reflects changes to the contents of an object, not its metadata.<br>Type: String<br>Ancestor: PostResponse |
| <i>Key</i>      | 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.<br>Type: String<br>Ancestor: PostResponse  |
| <i>Location</i> | URI of the object.<br>Type: String<br>Ancestor: PostResponse   |

### 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 \(p. 3\)](#).

## Examples

### Sample Request

```
POST /Neo HTTP/1.1
Content-Length: 4
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Expect: the 100-continue HTTP status code
```

*ObjectContent*

### Sample Response With Versioning Suspended

The following shows a sample response when bucket versioning is suspended.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEbla0wiIOIx1+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
```

Notice in this response the version ID is `null`.

## Sample Response With Versioning Enabled

The following shows a sample response when bucket versioning is enabled.

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIx1+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fjj3HHNVfdsQUIFDNsidf038jfdsjGFDSIRp
Date: Wed, 01 Mar 2009 12:00:00 GMT
ETag: "828ef3fdfa96f00ad9f27c383fc9ac7f"
Content-Length: 0
Connection: close
Server: AmazonS3
```

## Related Resources

- [PUT Object - Copy \(p. 142\)](#)
- [POST Object \(p. 122\)](#)
- [GET Object \(p. 106\)](#)

# 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 Amazon S3 receives multiple write requests for the same object simultaneously, all but the last object written will be overwritten. 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 the Content-MD5 header, 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 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 prior to 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 (e.g., authentication failure or redirect). For more information on 100-continue HTTP status code, go to Section 8.2.3 of <http://www.ietf.org/rfc/rfc2616.txt>.

## 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 \(p. 59\)](#).

If you enable versioning on a bucket, when Amazon S3 receives multiple write requests for the same object simultaneously, all of the objects will be stored.

To see sample requests that use versioning, see [Sample Request \(p. 133\)](#).

## Reduced Redundancy Storage

RRS enables customers to reduce their costs by storing non-critical, reproducible data at lower levels of redundancy than Amazon S3's standard storage. RRS provides a cost-effective, highly available solution for distributing or sharing content that is durably stored elsewhere, or for storing thumbnails, transcoded media, or other processed data that can be easily reproduced. The RRS option stores objects on multiple devices across multiple facilities, providing 400 times the durability of a typical disk drive, but does not replicate objects as many times as standard Amazon S3 storage, and thus is even more cost effective.

To store an object using reduced redundancy, set the `x-amz-storage-class` request header to `REDUCED_REDUNDANCY`. The default value is `STANDARD`.

## Requests

### Syntax

```
PUT /ObjectName HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### 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. For more information, see [Common Request Headers \(p. 12\)](#).

| Name                       | Description   | Required |
|----------------------------|---|----------|
| <i>Cache-Control</i>       | 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> .<br>Type: String<br>Default: None<br>Constraints: None  | No       |
| <i>Content-Disposition</i> | 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> .<br>Type: String<br>Default: None<br>Constraints: None  | No       |
| <i>Content-Encoding</i>    | 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 <i>Content-Type</i> 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> .<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>Content-Length</i>      | 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> .<br>Type: String<br>Default: None<br>Constraints: None   | Yes      |

**Amazon Simple Storage Service API Reference**  
**PUT Object**

---

| Name                | Description   | Required |
|---------------------|---|----------|
| <i>Content-MD5</i>  | <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, go to <a href="#">REST Authentication</a> in the <a href="#">Amazon Simple Storage Service Developer Guide</a></p> <p>Type: String<br/>Default: None<br/>Constraints: None</p> | No       |
| <i>Content-Type</i> | <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<br/>Default: binary/octet-stream<br/>Valid Values: MIME types<br/>Constraints: None</p>   | No       |
| <i>Expect</i>       | <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.</p> <p>Type: String<br/>Default: None<br/>Valid Values: 100-continue<br/>Constraints: None</p>   | No       |
| <i>Expires</i>      | <p>Number of milliseconds before expiration</p> <p>Type: Int<br/>Default: None<br/>Constraints: None</p>  | No       |
| <i>x-amz-acl</i>    | <p>The canned ACL to apply to the object. For more information, go to <a href="#">REST Access Policy</a> in the <a href="#">Amazon Simple Storage Service Developer Guide</a></p> <p>Type: String<br/>Default: private<br/>Valid Values: private   public-read   public-read-write   authenticated-read   bucket-owner-read   bucket-owner-full-control<br/>Constraints: None</p>   | No       |

| Name                                | Description  | Required |
|-------------------------------------|--|----------|
| <i>x-amz-meta-</i>                  | Any header starting with this prefix is considered user metadata. It will be stored with the object and returned when you retrieve the object. The PUT request header is limited to 8 KB in size. Within the PUT request header, the user-defined metadata is limited to 2 KB in size. User-defined metadata a set of key-value pairs. The size of user-defined metadata is measured by taking the sum of the number of bytes in the UTF-8 encoding of each key and value.<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>x-amz-server-side-encryption</i> | Specifies server-side encryption algorithm to use when Amazon S3 creates an object.<br>Type: String<br>Valid Value: <code>AES256</code>  | No       |
| <i>x-amz-storage-class</i>          | RRS enables customers to reduce their costs by storing non-critical, reproducible data at lower levels of redundancy than Amazon S3's standard storage.<br>Type: Enum<br>Default: <code>STANDARD</code><br>Valid Values: <code>STANDARD</code>   <code>REDUCED_REDUNDANCY</code><br>Constraints: None  | No       |

## 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 \(p. 14\)](#).

| Name                                | Description  |
|-------------------------------------|--|
| <i>x-amz-server-side-encryption</i> | If you request server-side encryption when adding an object, the response includes this header confirming the encryption algorithm used.<br>Type: String |
| <i>x-amz-version-id</i>             | Version of the object.<br>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 \(p. 3\)](#).



## Examples

### Sample Request

The following request stores the image, `my-image.jpg`, in the bucket, `myBucket`.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 11434
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/xUQxMsF7xiEb1a0wiIOIxl+zbwZ163pt7
x-amz-request-id: 0A49CE4060975EAC
Date: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: AmazonS3
```

### Sample Response With Versioning Enabled

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiEb1a0wiIOIxl+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
```

Note that this response contains the response header `x-amz-version-id`.

### Sample Request Using Reduced Redundancy

The following request stores the image, `my-image.jpg`, in the bucket, `myBucket`.

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
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: LriYPLdmOdAiIfgSm/FlYsViT1LW94/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
```

## Related Resources

- [PUT Object - Copy \(p. 142\)](#)
- [POST Object \(p. 122\)](#)
- [GET Object \(p. 106\)](#)

# 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. (To set the ACL of an object when you put it into a bucket, use the `x-amz-acl` request header.) You must have `WRITE_ACP` permission to set the ACL of an object.

## Versioning

The ACL of an object is set at the object version level. By default, `PUT` sets the ACL of the latest 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. 140).

## Requests

### Syntax

```
PUT /ObjectName?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue

<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

| Name                       | Description  | Required |
|----------------------------|--|----------|
| <i>Cache-Control</i>       | 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> .<br>Type: String<br>Default: None<br>Constraints: None   | No       |
| <i>Content-Disposition</i> | 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> .<br>Type: String<br>Default: None<br>Constraints: None   | No       |
| <i>Content-Encoding</i>    | 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 <i>Content-Type</i> 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> .<br>Type: String<br>Default: None<br>Constraints: None  | No       |
| <i>Content-Length</i>      | 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> .<br>Type: String<br>Default: None<br>Constraints: None  | Yes      |
| <i>Content-MD5</i>         | 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 <a href="#">Amazon Simple Storage Service Developer Guide</a> .<br>Type: String<br>Default: None<br>Constraints: None | No       |
| <i>Content-Type</i>        | 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> .<br>Type: String<br>Default: binary/octet-stream<br>Valid Values: MIME types<br>Constraints: None   | No       |

| Name             | Description   | Required |
|------------------|---|----------|
| <i>Expect</i>    | 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.<br>Type: String<br>Default: None<br>Valid Values: 100-continue<br>Constraints: None   | No       |
| <i>Expires</i>   | Number of milliseconds before expiration<br>Type: Int<br>Default: None<br>Constraints: None   | No       |
| <i>x-amz-acl</i> | The canned ACL to apply to the object. For more information, go to <a href="#">REST Access Policy</a> in the <a href="#">Amazon Simple Storage Service Developer Guide</a> .<br>Type: String<br>Default: private<br>Valid Values: private   public-read   public-read-write   authenticated-read   bucket-owner-read   bucket-owner-full-control<br>Constraints: None | No       |

## Request Elements

You use the following parameters in the body of a `GET` request to set ACL permissions for a grantee.

| Name                       | Description   |
|----------------------------|---|
| <i>AccessControlList</i>   | Container for ACL information<br>Type: Container<br>Ancestors: AccessControlPolicy  |
| <i>AccessControlPolicy</i> | Contains the elements that set the ACL permissions for an object per Grantee<br>Type: Container<br>Ancestors: None          |
| <i>DisplayName</i>         | Screen name of the bucket owner<br>Type: String<br>Ancestors: AccessControlPolicy.Owner                                     |
| <i>Grant</i>               | Container for the grantee and his or her permissions<br>Type: Container<br>Ancestors: AccessControlPolicy.AccessControlList |

| Name              | Description   |
|-------------------|---|
| <i>Grantee</i>    | The subject whose permissions are being set.<br>Type: String<br>Valid Values: DisplayName   EmailAddress   AuthenticatedUser. For more information, see <a href="#">Grantee Values (p. 138)</a> .<br>Ancestors: AccessControlPolicy.AccessControlList.Grant |
| <i>ID</i>         | ID of the bucket owner, or the ID of the grantee<br>Type: String<br>Ancestors: AccessControlPolicy.Owner or AccessControlPolicy.AccessControlList.Grant   |
| <i>Owner</i>      | Container for the bucket owner's display name and ID<br>Type: Container<br>Ancestors: AccessControlPolicy   |
| <i>Permission</i> | Specifies the permission given to the grantee<br>Type: String<br>Valid Values: FULL_CONTROL   WRITE   WRITE_ACP   READ   READ_ACP<br>Ancestors: AccessControlPolicy.AccessControlList.Grant   |

### Grantee Values

You can specify the person (grantee) that you're assigning access rights to (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>ID</ID>
  <DisplayName>GranteesEmail</DisplayName>
</Grantee>
```

*DisplayName* is optional and ignored in the request.

- By E-mail address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="AmazonCustomerByEmail">
  <EmailAddress>Grantees@email.com</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>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</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 \(p. 14\)](#).

| Name             | Description  |
|------------------|--|
| x-amz-version-id | Version of the object whose ACL is being set.<br>Type: String<br>Default: None |

### 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 \(p. 3\)](#).

## Examples

### Sample Request

The following request sets the ACL on the specified object.

```
PUT /my-image.jpg?acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>8a6925ce4adf588e97f21c32aa379004fef</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>8a6925ce4adf588a45379004fef</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: eftixk72aD6Ap5lT9ASled4OpIszj7UDNEHGran
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+rmSpXd3dIb
rHY+MTRCxf3vjVBH40NrjfkD HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>8a6925ce4adf5f21c32aa379004fef</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
        <ID>8a6925ce4adf588a4532142d3f74dd8c71fa124blddee97f21c32aa379004fef</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: eftixk72aD6Ap5lu8yU9ASled4OpIszj7UDNEHGran
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
```

## Related Resources

- [PUT Object - Copy \(p. 142\)](#)
- [POST Object \(p. 122\)](#)



- [GET Object \(p. 106\)](#)

## 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 API. For conceptual information on multipart upload, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon S3 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. To override the default ACL setting, use the `x-amz-acl` header to specify a new ACL when generating a copy request. For more information, see [Amazon S3 ACLs](#).

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 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 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 prefixed with `x-amz-` must be signed, including `x-amz-copy-source`.

There are two opportunities for a copy request to return an error. One can occur when Amazon S3 receives the copy request and the other can occur 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 response. This means that a 200 response can contain either a success or an error. Make sure to design your application to parse the contents of the response and handle it appropriately.

If the copy is successful, you receive a response that contains the information about the copied object.



#### Note

If the request is an HTTP 1.1 request, the response is chunk encoded. Otherwise, it will not contain the content-length and you will need to read the entire body.

### Versioning

By default, `x-amz-copy-source` identifies the latest version of an object to copy. (If the latest 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 Amazon S3 generates is always `null`.

To see sample requests that use Versioning, see [Sample Request Copying a Specified Version of an Object](#) (p. 147).

## 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: signatureValue
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. For more information, see [Common Request Headers](#) (p. 12).

| Name             | Description   | Required |
|------------------|---|----------|
| <i>x-amz-acl</i> | The canned ACL to apply to the object. For more information, go to <a href="#">REST Access Policy</a> in the <a href="#">Amazon Simple Storage Service Developer Guide</a> .<br>Type: String<br>Default: private<br>Valid Values: private   public-read   public-read-write   authenticated-read   bucket-owner-read   bucket-owner-full-control<br>Constraints: None | No       |

| Name                                   | Description   | Required |
|--|---|----------|
| <i>x-amz-copy-source</i>               | <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: 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>   | Yes      |
| <i>x-amz-metadata-directive</i>        | <p>Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request. If copied, the metadata, except for the version ID, remains unchanged. Otherwise, all original metadata is replaced by the metadata you specify.</p> <p>Type: String</p> <p>Default: <code>COPY</code></p> <p>Valid values: <code>COPY</code>   <code>REPLACE</code></p> <p>Constraints: Values other than <code>COPY</code> or <code>REPLACE</code> result in an immediate 400-based error response. You cannot copy an object to itself unless the <code>MetadataDirective</code> header is specified and its value set to <code>REPLACE</code>.</p> <p>For information on supported metadata, see <a href="#">Common Request Headers (p. 12)</a></p> | No       |
| <i>x-amz-copy-source-if-match</i>      | <p>Copies the object if its entity tag (ETag) matches the specified tag; otherwise, the request returns a 412 HTTP status code error (precondition failed).</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 cannot be used with other conditional copy headers.</p>   | No       |
| <i>x-amz-copy-source-if-none-match</i> | <p>Copies the object if its entity tag (ETag) is different than the specified ETag; otherwise, the request returns a 412 HTTP status code error (failed condition).</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 cannot be used with other conditional copy headers.</p>   | No       |

| Name   | Description   | Required |
|--|---|----------|
| <i>x-amz-copy-source-if-unmodified-since</i> | Copies the object if it hasn't been modified since the specified time; otherwise, the request returns a 412 HTTP status code error (precondition failed).<br>Type: String<br>Default: None<br>Constraints: This must be a valid HTTP date. For more information, go to <a href="http://www.ietf.org/rfc/rfc2616.txt">http://www.ietf.org/rfc/rfc2616.txt</a> . This header can be used with <i>x-amz-copy-source-if-match</i> , but cannot be used with other conditional copy headers. | No       |
| <i>x-amz-copy-source-if-modified-since</i>   | Copies the object if it has been modified since the specified time; otherwise, the request returns a 412 HTTP status code error (failed condition).<br>Type: String<br>Default: None<br>Constraints: This must be a valid HTTP date. This header can be used with <i>x-amz-copy-source-if-none-match</i> , but cannot be used with other conditional copy headers.  | No       |
| <i>x-amz-server-side-encryption</i>          | Specifies the server-side encryption algorithm to use when Amazon S3 creates the target object.<br>Type: String<br>Valid Value: AES256  | No       |
| <i>x-amz-storage-class</i>                   | RRS enables customers to reduce their costs by storing non-critical, reproducible data at lower levels of redundancy than Amazon S3's standard storage.<br>Type: Enum<br>Default: STANDARD<br>Valid Values: STANDARD   REDUCED_REDUNDANCY<br>Constraints: None  | 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 \(p. 14\)](#).

| Name                                | Description  |
|-------------------------------------|--|
| <i>x-amz-copy-source-version-id</i> | Version of the source object that was copied.<br>Type: String  |
| <i>x-amz-server-side-encryption</i> | If you request server-side encryption, the response includes this header confirming the encryption algorithm used for the target object.<br>Type: String |
| <i>x-amz-version-id</i>             | Version of the copied object in the destination bucket.<br>Type: String  |

## Response Elements

| Name                    | Description  |
|-------------------------|--|
| <i>CopyObjectResult</i> | Container for all response elements.<br>Type: Container<br>Ancestor: None  |
| <i>ETag</i>             | Returns the ETag of the new object. The ETag only reflects changes to the contents of an object, not its metadata.<br>Type: String<br>Ancestor: CopyObjectResult |
| <i>LastModified</i>     | Returns the date the object was last modified.<br>Type: String<br>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 \(p. 3\)](#).

## 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: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## 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+rmSpXd3dIb
rHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
x-amz-version-id: QUpfdndhfd8438MNFDN93jdnJFkdmqnh893
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 and  
*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 key, *my-image.jpg*, with the specified version ID and copies it into the bucket, *bucket*, and gives it the key, *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?versionId=3/L4kqtJlcpXroDTDmJ+rmSpXd3dIb
rHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

## 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. Note that 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+rmSpXd3dIb
rHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
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>
```

## Related Resources

- [Copying Objects](#)
- [PUT Object \(p. 129\)](#)
- [GET Object \(p. 106\)](#)



# Initiate Multipart Upload

## Description

This operation initiates a multipart upload and returns an upload ID. This upload ID is used to associate all the parts in the specific multipart upload. You specify this upload ID in each of your subsequent upload part requests (see [Upload Part \(p. 153\)](#)). You also include this upload ID in the final request to either complete or abort the multipart upload request.

For more information on multipart uploads, go to [Multipart Upload Overview](#) in the *Amazon S3 Developer Guide*.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer Guide*.



### Note

If you create an object using the multipart upload APIs, currently you cannot copy the object between regions.

## Requests

### Syntax

```
POST /ObjectName?uploads HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
```

### Request Parameters

This operation does not use request parameters.

### Request Headers

| Name                       | Description   | Required |
|----------------------------|---|----------|
| <i>Cache-Control</i>       | 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> .<br>Type: String<br>Default: None | No       |
| <i>Content-Disposition</i> | 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> .<br>Type: String<br>Default: None               | No       |

| Name                                | Description  | Required |
|-------------------------------------|--|----------|
| <i>Content-Encoding</i>             | 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 <i>Content-Type</i> 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> .<br>Type: String<br>Default: None | No       |
| <i>Content-Type</i>                 | A standard MIME type describing the format of the object data. 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> .<br>Type: String<br>Default: binary/octet-stream<br>Constraints: MIME types only   | No       |
| <i>Expires</i>                      | The date and time at which the object is no longer cacheable. 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> .<br>Type: String<br>Default: None   | No       |
| <i>x-amz-acl</i>                    | The canned ACL to apply, to the object that is created after completing multipart upload. For more conceptual information, go to <a href="#">Canned ACL</a> in the <i>Amazon S3 Developer Guide</i> .<br>Type: String<br>Default: private<br>Valid Values: private   public-read   public-read-write   authenticated-read   bucket-owner-read   bucket-owner-full-control                                    | No       |
| <i>x-amz-meta-</i>                  | Any header starting with this prefix is considered user metadata. It will be stored with the object and returned when you retrieve the object.<br>Type: String<br>Default: None  | No       |
| <i>x-amz-server-side-encryption</i> | Specifies the server-side encryption algorithm to use. As you uploads individual object parts, Amazon S3 applies server-side encryption to each part you upload.<br>Type: String<br>Valid Value: AES256  | No       |
| <i>x-amz-storage-class</i>          | The type of storage to use, for the object that is created after successful multipart upload.<br>Type: String<br>Valid Values: STANDARD   REDUCED_REDUNDANCY<br>Default: STANDARD  | No       |

## 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 \(p. 14\)](#).

| Name                                | Description  |
|-------------------------------------|--|
| <i>x-amz-server-side-encryption</i> | If you specify server-side encryption in your request, the response includes this header. It confirms the encryption algorithm that will be used for the object that is created after successful multipart upload.<br>Type: String |

### Response Elements

| Name                          | Description   |
|-------------------------------|---|
| InitiateMultipartUploadResult | Container for response.<br>Type: Container<br>Children: Bucket, Key, UploadId<br>Ancestors: None                            |
| Bucket                        | Name of the bucket to which the multipart upload was initiated.<br>Type: string<br>Ancestors: InitiateMultipartUploadResult |
| Key                           | Object key for which the multipart upload was initiated.<br>Type: String<br>Ancestors: InitiateMultipartUploadResult        |
| UploadId                      | ID for the initiated multipart upload.<br>Type: String<br>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 \(p. 3\)](#).

## 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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZCBieSB1bHZpbmc=
```

## 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>VXBsb2FkIElEIGZvcia2aWWpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA</UploadId>
</InitiateMultipartUploadResult>
```

## Related Actions

- [Upload Part \(p. 153\)](#)
- [Complete Multipart Upload \(p. 161\)](#)
- [Abort Multipart Upload \(p. 166\)](#)
- [List Parts \(p. 168\)](#)
- [List Multipart Uploads \(p. 65\)](#)

# 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 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 \(p. 156\)](#).

You must initiate a multipart upload (see [Initiate Multipart Upload \(p. 149\)](#)) 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.

For more information on multipart uploads, go to [Multipart Upload Overview](#) in the *Amazon S3 Developer guide*.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer guide*.

## Requests

### Syntax

```
PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Content-Length: Size
Authorization: Signature
```

### 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. For more information, see [Common Request Headers \(p. 12\)](#).

| Name                  | Description  | Required |
|-----------------------|--|----------|
| <i>Content-Length</i> | 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> .<br>Type: Integer<br>Default: None  | Yes      |
| <i>Content-MD5</i>    | 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> .<br>Type: String<br>Default: None | No       |
| <i>Expect</i>         | 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> .<br>Type: String<br>Default: None<br>Valid Values: 100-continue  | No       |

## 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 \(p. 14\)](#).

| Name                                | Description  |
|-------------------------------------|--|
| <i>x-amz-server-side-encryption</i> | If you specified server-side encryption in your initiate multipart upload request, the response includes this header. It confirms the encryption algorithm that Amazon S3 used to encrypt the part you uploaded.<br>Type: String |

### 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 \(p. 3\)](#).

## 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=VCVsb2FkIElEIGZvciBlbZZpbm
cncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR 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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZGgieSRlbHZZpbmc=

***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: Vvag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/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
```

## Related Actions

- [Initiate Multipart Upload \(p. 149\)](#)
- [Complete Multipart Upload \(p. 161\)](#)
- [Abort Multipart Upload \(p. 166\)](#)
- [List Parts \(p. 168\)](#)
- [List Multipart Uploads \(p. 65\)](#)

# 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.



### 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 \(p. 153\)](#).

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 conceptual information on multipart uploads, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon S3 Developer guide*. For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer guide*. To more information about copying objects using a single atomic operation vs. the multipart upload, go to [Operations On Objects](#) in the *Amazon S3 Developer guide*.

## 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: Signature
```

### 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. For more information, see [Common Request Headers \(p. 12\)](#).

| Name                     | Description   | Required |
|--------------------------|---|----------|
| <i>x-amz-copy-source</i> | The name of the source bucket and the source object key name separated by a slash (/).<br>Type: String<br>Default: None | Yes      |



| Name                           | Description   | Required |
|--------------------------------|---|----------|
| <i>x-amz-copy-source-range</i> | <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>You can copy a range only if the source object is greater than 5 GB.</p> <p>This request header is not required when copying an entire source object.</p> <p>Type: Integer<br/>Default: None</p> | No       |

The following conditional headers are based on the object that the *x-amz-copy-source* header specifies.

| Name   | Description  | Required |
|--|--|----------|
| <i>x-amz-copy-source-if-match</i>            | <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 412 <i>precondition failed</i> error.</p> <p>Type: String<br/>Default: None</p>                     | No       |
| <i>x-amz-copy-source-if-none-match</i>       | <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 412 <i>precondition failed</i> error.</p> <p>Type: String<br/>Default: None</p> | No       |
| <i>x-amz-copy-source-if-unmodified-since</i> | <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 412 <i>precondition failed</i> error.</p> <p>Type: String<br/>Default: None</p>   | No       |
| <i>x-amz-copy-source-if-modified-since</i>   | <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 412 <i>precondition failed</i> error.</p> <p>Type: String<br/>Default: None</p>   | No       |

## 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 latest version of the object to copy. If the latest 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 response. For more information, see [Common Response Headers](#) (p. 14).

| Name                                | Description   |
|-------------------------------------|---|
| <i>x-amz-copy-source-version-id</i> | The version of the source object that was copied, if you have enabled versioning on the source bucket.<br>Type: String  |
| <i>x-amz-server-side-encryption</i> | If you specified server-side encryption 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.<br>Type: String |

### Response Elements

| Name                  | Description   |
|-----------------------|---|
| <i>CopyPartResult</i> | Container for all response elements.<br>Type: Container<br>Ancestor: None                       |
| <i>ETag</i>           | Returns the ETag of the new part.<br>Type: String<br>Ancestor: <i>CopyPartResult</i>            |
| <i>LastModified</i>   | Returns the date the part was last modified.<br>Type: String<br>Ancestor: <i>CopyPartResult</i> |

## 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 \(p. 3\)](#).

## 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=VCVsb2FkIEElEIGZvciBlbZZpbm
cncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR 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: AWS VGhpcyBtZXNzYWdlIHNoZ251ZGgieSRlbHZZpbmc=
```

### 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>2009-10-28T22:32:00</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=VCVsb2FkIElEIGZvciBlbZZpbm
cncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR 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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZGgieSRlbHZpbmc=
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=VCVsb2FkIElEIGZvciBlbZZpbm
cncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR 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+rm
SpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
x-amz-copy-source-range: bytes=500-6291456
Authorization: AWS VGhpcyBtZXNzYWdlIHNPZ25lZGgieSRlbHZpbmc=
```

## 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+rmSpXd3dIb
rHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
Date: Mon, 11 Apr 2011 20:34:56 GMT
Server: AmazonS3

<CopyPartResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyPartResult>
```

## Related Actions

- [Initiate Multipart Upload \(p. 149\)](#)
- [Upload Part \(p. 153\)](#)
- [Complete Multipart Upload \(p. 161\)](#)
- [Abort Multipart Upload \(p. 166\)](#)
- [List Parts \(p. 168\)](#)
- [List Multipart Uploads \(p. 65\)](#)

# 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 \(p. 153\)](#)). 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. 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 S3 Developer guide*.

For more information on multipart uploads, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon S3 Developer guide*.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer guide*.

## Requests

### Syntax

```
POST /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Content-Length: Size
Authorization: Signature

<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 \(p. 12\)](#)

## Request Elements

| Name                           | Description  | Required |
|--------------------------------|--|----------|
| <i>CompleteMultipartUpload</i> | Container for the request.<br><br>Ancestor: None<br>Type: Container<br>Children: One or more <i>Part</i> elements  | Yes      |
| <i>Part</i>                    | Container for elements related to a particular previously uploaded part.<br><br>Ancestor: <i>CompleteMultipartUpload</i><br>Type: Container<br>Children: <i>PartNumber</i> , <i>ETag</i> | Yes      |
| <i>PartNumber</i>              | Part number that identifies the part.<br><br>Ancestor: <i>Part</i><br>Type: Integer  | Yes      |
| <i>ETag</i>                    | Entity tag returned when the part was uploaded.<br><br>Ancestor: <i>Part</i><br>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 \(p. 14\)](#).

| Header                              | Description  |
|-------------------------------------|--|
| <i>x-amz-server-side-encryption</i> | If you specified server-side encryption in your initiate multipart upload request, the response includes this header confirming the encryption algorithm Amazon S3 used to save your object data to disks in its data centers.<br><br>Type: String |
| <i>x-amz-version-id</i>             | Version ID of the newly created object, in case the bucket has versioning turned on.<br><br>Type: String   |

## Response Elements

| Name                                       | Description  |
|--|--|
| <code>CompleteMultipartUploadResult</code> | Container for the response<br>Type: Container<br>Children: <i>Location</i> , <i>Bucket</i> , <i>Key</i> , <i>ETag</i><br>Ancestors: None   |
| <code>Location</code>                      | The URI that identifies the newly created object.<br>Type: URI<br>Ancestors: <i>CompleteMultipartUploadResult</i>  |
| <code>Bucket</code>                        | The name of the bucket that contains the newly created object.<br>Type: String<br>Ancestors: <i>CompleteMultipartUploadResult</i>  |
| <code>Key</code>                           | The object key of the newly created object.<br>Type: String<br>Ancestors: <i>CompleteMultipartUploadResult</i>   |
| <code>ETag</code>                          | 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.<br>Type: String<br>Ancestors: <i>CompleteMultipartUploadResult</i> |

## Special Errors

| Error Code                    | Description   | HTTP Status Code |
|-------------------------------|---|------------------|
| <code>InvalidPart</code>      | 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  |
| <code>InvalidPartOrder</code> | The list of parts was not in ascending order. Parts list must be specified in order by part number.   | 400 Bad Request  |
| <code>NoSuchUpload</code>     | 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 \(p. 3\)](#).

## Examples

### Sample Request

The following Complete Multipart Upload request specifies three parts in the *CompleteMultipartUpload* element.

```
POST /example-object?uploadId=AAAsb2FkIElEIGZvciBlbHZpbmcncyWeeS1tb3ZpZS5tMnRzIR
RwbG9hZA HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Content-Length: 391
Authorization: AWS AAbbcyBtZXNzYWdlRRRpZ25lZCBieSB1bHZpbmc=

<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"a54357aff0632cce46d942af68356b38"</ETag>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <ETag>"0c78aef83f66abc1fale8477f296d394"</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: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/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: Uuag1LuByRx9e6j5Onimru9pO4ZVKnJ2Qz7/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>InternalServerError</Code>
  <Message>We encountered an internal error. Please try again.</Message>
  <RequestId>656c76696e6727732072657175657374</RequestId>
  <HostId>Uuag1LuByRx9e6j5Onimru9p04ZVKnJ2Qz7/C1NPcfTWAtRPfTaOfg==</HostId>
</Error>
```

## Related Actions

- [Initiate Multipart Upload \(p. 149\)](#)
- [Upload Part \(p. 153\)](#)
- [Abort Multipart Upload \(p. 166\)](#)
- [List Parts \(p. 168\)](#)
- [List Multipart Uploads \(p. 65\)](#)

# 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.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer Guide*.

## Requests

### Syntax

```
DELETE /ObjectName?uploadId=UploadId HTTP/1.1  
Host: BucketName.s3.amazonaws.com  
Date: Date  
Authorization: Signature
```

### 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. 12).

### 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. 14).

### 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 \(p. 3\)](#).

## Examples

### Sample Request

The following request aborts a multipart upload identified by its upload ID.

```
DELETE /example-object?uploadId=VXBsb2FkIElEIGZvcjBlbHZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZ HTTP/1.1
Host: example-bucket.s3.amazonaws.com
Date: Mon, 1 Nov 2010 20:34:56 GMT
Authorization: AWS QQxxcyBtZXNzYWdlIHNPZ25lZCBieSB1bHZpabc=
```

### 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 \(p. 149\)](#)
- [Upload Part \(p. 153\)](#)
- [Complete Multipart Upload \(p. 161\)](#)
- [List Parts \(p. 168\)](#)
- [List Multipart Uploads \(p. 65\)](#)

# 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 \(p. 149\)](#)). 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, go to [Uploading Objects Using Multipart Upload](#) in the *Amazon S3 Developer Guide*.

For information on permissions required to use the multipart upload API, go to [Multipart Upload API and Permissions](#) in the *Amazon S3 Developer Guide*.

## Requests

### Syntax

```
GET /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: Date
Authorization: Signature
```

### 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 |
|---------------------------|---|----------|
| <i>uploadId</i>           | Upload ID identifying the multipart upload whose parts are being listed.<br>Type: String<br>Default: None                                 | Yes      |
| <i>max-parts</i>          | Sets the maximum number of parts to return in the response body.<br>Type: String<br>Default: 1,000  | No       |
| <i>part-number-marker</i> | Specifies the part after which listing should begin. Only parts with higher part numbers will be listed.<br>Type: String<br>Default: None | No       |

### Request Headers

This operation uses only request headers common to most requests. For more information, see [Common Request Headers \(p. 12\)](#).

## 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. 14).

### Response Elements

| Name            | Description   |
|-----------------|---|
| ListPartsResult | Container for the response.<br>Children: <i>Bucket, Key, UploadId, Initiator, Owner, StorageClass, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part</i><br>Type: Container   |
| Bucket          | Name of the bucket to which the multipart upload was initiated.<br>Type: String<br>Ancestor: <i>ListPartsResult</i>   |
| Key             | Object key for which the multipart upload was initiated.<br>Type: String<br>Ancestor: <i>ListPartsResult</i>  |
| UploadId        | Upload ID identifying the multipart upload whose parts are being listed.<br>Type: String<br>Ancestor: <i>ListPartsResult</i>  |
| Initiator       | Container element that identifies who initiated the multipart upload. If the initiator is an AWS account, this element provides the same information as the <i>Owner</i> element. If the initiator is an IAM User, then this element provides the user ARN and display name.<br>Children: <i>ID, DisplayName</i><br>Type: Container<br>Ancestor: <i>ListPartsResult</i> |
| ID              | 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.<br>Type: String<br>Ancestor: <i>Initiator</i>   |
| DisplayName     | Principal's name.<br>Type: String<br>Ancestor: <i>Initiator</i>   |

| Name                 | Description  |
|----------------------|--|
| Owner                | 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.<br>Children: <i>ID</i> , <i>DisplayName</i><br>Type: Container<br>Ancestor: <i>ListPartsResult</i> |
| StorageClass         | Class of storage (STANDARD or REDUCED_REDUNDANCY) used to store the uploaded object.<br>Type: String<br>Ancestor: <i>ListPartsResult</i>   |
| PartNumberMarker     | Part number after which listing begins.<br>Type: Integer<br>Ancestor: <i>ListPartsResult</i>   |
| NextPartNumberMarker | When a list is truncated, this element specifies the last part in the list, as well as the value to use for the <i>part-number-marker</i> request parameter in a subsequent request.<br>Type: Integer<br>Ancestor: <i>ListPartsResult</i>  |
| MaxParts             | Maximum number of parts that were allowed in the response.<br>Type: Integer<br>Ancestor: <i>ListPartsResult</i>  |
| IsTruncated          | Indicates whether the returned list of parts is truncated. A <i>true</i> value indicates that the list was truncated. A list can be truncated if the number of parts exceeds the limit returned in the <i>MaxParts</i> element.<br>Type: Boolean<br>Ancestor: <i>ListPartsResult</i>               |
| Part                 | Container for elements related to a particular part. A response can contain zero or more <i>Part</i> elements.<br>Children: <i>PartNumber</i> , <i>LastModified</i> , <i>ETag</i> , <i>Size</i><br>Type: String<br>Ancestor: <i>ListPartsResult</i>  |
| PartNumber           | Part number identifying the part.<br>Type: Integer<br>Ancestor: <i>Part</i>  |
| LastModified         | Date and time at which the part was uploaded.<br>Type: Date<br>Ancestor: <i>Part</i>   |
| ETag                 | Entity tag returned when the part was uploaded.<br>Type: String<br>Ancestor: <i>Part</i>   |

| Name | Description   |
|------|---|
| Size | Size of the uploaded part data.<br>Type: Integer<br>Ancestor: <i>Part</i> |

## 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 \(p. 3\)](#).

## 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=XXBsb2FkIElEIGZvciblbHZpbmcncyVcdS1tb3ZpZS5tMnRzEEEw
bG9hZA&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: AWS VGhpcyBtZXNzYWdlIHNPZ25lZCBieSB1bHZpbmc=
```

### 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>XXBsb2FkIElEIGZvcilBlbHZpbmcncyVcdS1tb3ZpZS5tMnRzEEEwbG9hZA</UploadId>

<Initiator>
  <ID>arn:aws:iam::111111111111:user/some-user-11116a31-17b5-4fb7-9df5-
b288870f11xx</ID>
  <DisplayName>umat-user-11116a31-17b5-4fb7-9df5-b288870f11xx</DisplayName>
</Initiator>
<Owner>
  <ID>x1x16700c70b0b05597d7ecd6a3f92be</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>"7778aef83f66abclfale8477f296d394"</ETag>
  <Size>10485760</Size>
</Part>
<Part>
  <PartNumber>3</PartNumber>
  <LastModified>2010-11-10T20:48:33.000Z</LastModified>
  <ETag>"aaaal8db4cc2f85cedef654fccc4a4x8"</ETag>
  <Size>10485760</Size>
</Part>
</ListPartsResult>
```

## Related Actions

- [Initiate Multipart Upload \(p. 149\)](#)
- [Upload Part \(p. 153\)](#)
- [Complete Multipart Upload \(p. 161\)](#)
- [Abort Multipart Upload \(p. 166\)](#)
- [List Multipart Uploads \(p. 65\)](#)



# SOAP API

---

## Topics

- [Operations on the Service \(p. 173\)](#)
- [Operations on Buckets \(p. 174\)](#)
- [Operations on Objects \(p. 183\)](#)

This section describes the SOAP API with respect to service, bucket, and object operations.



## Note

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.

## Operations on the Service

## Topics

- [ListAllMyBuckets \(p. 173\)](#)

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

## ListAllMyBuckets

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>1D9FVRAYCP1VJEXAMPLE=</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>bcaf1ffd86f411b61ca5fb16fd081034f</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

### Topics

- [CreateBucket](#) (p. 175)

- [DeleteBucket](#) (p. 176)
- [ListBucket](#) (p. 176)
- [GetBucketAccessControlPolicy](#) (p. 179)
- [SetBucketAccessControlPolicy](#) (p. 180)
- [GetBucketLoggingStatus](#) (p. 181)
- [SetBucketLoggingStatus](#) (p. 182)

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

## CreateBucket

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>1D9FVRAYCP1VJEXAMPLE=</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](#) (p. 176)

## DeleteBucket

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> 1D9FVRAYCP1VJEXAMPLE=</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

The `ListBucket` operation returns information about some of the items in the bucket.

For a general introduction to the list operation, see the [Listing 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>1D9FVRAYCP1VJEXAMPLE=</AWSAccessKeyId>
  <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListBucket>
```

### Parameters

| Name             | Description  | Required |
|------------------|--|----------|
| <i>prefix</i>    | 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.<br>Type: String<br>Default: None  | No       |
| <i>marker</i>    | 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.<br>Type: String<br>Default: None     | No       |
| <i>max-keys</i>  | 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.<br>Type: String<br>Default: None   | No       |
| <i>delimiter</i> | 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.<br>Type: String<br>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>"828ef3fd9a96f00ad9f27c383fc9ac7f"</ETag>
    <Size>5126</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>bcaflffd86f41ce161ca5fb16fd081034f</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  |
|-----------------------|--|
| <i>Contents</i>       | Metadata about each object returned.<br>Type: XML metadata<br>Ancestor: ListBucketResult   |
| <i>CommonPrefixes</i> | A response can contain <i>CommonPrefixes</i> only if you specify a <i>delimiter</i> . When you do, <i>CommonPrefixes</i> contains all (if there are any) keys between <i>Prefix</i> and the next occurrence of the string specified by <i>delimiter</i> . In effect, <i>CommonPrefixes</i> lists keys that act like subdirectories in the directory specified by <i>Prefix</i> . For example, if <i>prefix</i> is <i>notes/</i> and <i>delimiter</i> is a slash (/), in <i>notes/summer/july</i> , the common prefix is <i>notes/summer/</i> .<br>Type: String<br>Ancestor: ListBucketResult |

| Name               | Description   |
|--------------------|---|
| <i>Delimiter</i>   | 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.<br>Type: String<br>Ancestor: ListBucketResult |
| <i>IsTruncated</i> | 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 <i>MaxKeys</i> .<br>Type: String<br>Ancestor: boolean   |
| <i>Marker</i>      | Indicates where in the bucket to begin listing.<br>Type: String<br>Ancestor: ListBucketResult   |
| <i>MaxKeys</i>     | The maximum number of keys returned in the response body.<br>Type: String<br>Ancestor: ListBucketResult   |
| <i>Name</i>        | Name of the bucket.<br>Type: String<br>Ancestor: ListBucketResult   |
| <i>Prefix</i>      | Keys that begin with the indicated prefix.<br>Type: String<br>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

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>1D9FVRAYCP1VJEXAMPLE=</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>a9a7b886d6fd2441bf9b1c61be666e9</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

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>1D9FVRAYCP1VJEXAMPLE=</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



### Important

This document describes Beta functionality that is subject to change in future releases.

The `GetBucketLoggingStatus` retrieves the logging status for an existing bucket.

For a general introduction to this feature, see [Server Logs](#) . For information about the response document, see [Logging API](#) .

## Example

### 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>
      <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>
    </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>
      <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



### Important

This document describes Beta functionality that is subject to change in future releases.

The `SetBucketLoggingStatus` operation updates the logging status for an existing bucket.

For a general introduction to this feature, see [Server Logs](#) . For information about the response document, see [Logging API](#) .

### 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

### Topics

- [PutObjectInline](#) (p. 184)
- [PutObject](#) (p. 186)
- [CopyObject](#) (p. 189)
- [GetObject](#) (p. 193)
- [GetObjectExtended](#) (p. 198)
- [DeleteObject](#) (p. 198)
- [GetObjectAccessControlPolicy](#) (p. 199)
- [SetObjectAccessControlPolicy](#) (p. 200)

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

## PutObjectInline

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>1D9FVRAYCP1VJEXAMPLE=</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](#) (p. 186)
- [CopyObject](#) (p. 189)

## PutObject

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>1D9FVRAYCP1VJEXAMPLE=</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>"828ef3fdfa96f00ad9f27c383fc9ac7f"</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](#) (p. 189)



# CopyObject

## 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 [Amazon S3 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 |
|--------------------------|---|----------|
| <i>SourceBucket</i>      | The name of the source bucket.<br>Type: String<br>Default: None<br>Constraints: A valid source bucket.  | Yes      |
| <i>SourceKey</i>         | The key name of the source object.<br>Type: String<br>Default: None<br>Constraints: The key for a valid source object to which you have READ access.  | Yes      |
| <i>DestinationBucket</i> | The name of the destination bucket.<br>Type: String<br>Default: None<br>Constraints: You must have WRITE access to the destination bucket.  | Yes      |
| <i>DestinationKey</i>    | The key of the destination object.<br>Type: String<br>Default: None<br>Constraints: You must have WRITE access to the destination bucket.   | Yes      |
| <i>MetadataDirective</i> | Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.<br>Type: String<br>Default: COPY<br>Valid values: COPY   REPLACE<br>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. | No       |
| <i>Metadata</i>          | Specifies metadata name-value pairs to set for the object.If MetadataDirective is set to COPY, all metadata is ignored.<br>Type: String<br>Default: None<br>Constraints: None.  | No       |

| Name                               | Description   | Required |
|------------------------------------|---|----------|
| <i>AccessControlList</i>           | Grants access to users by e-mail addresses or canonical user ID.<br>Type: String<br>Default: None<br>Constraints: None  | No       |
| <i>CopySourceIfMatch</i>           | Copies the object if its entity tag (ETag) matches the specified tag; otherwise return a PreconditionFailed.<br>Type: String<br>Default: None<br>Constraints: None. If the Etag does not match, the object is not copied. | No       |
| <i>CopySourceIfNoneMatch</i>       | Copies the object if its entity tag (ETag) is different than the specified Etag; otherwise returns an error.<br>Type: String<br>Default: None<br>Constraints: None.   | No       |
| <i>CopySourceIfUnmodifiedSince</i> | Copies the object if it hasn't been modified since the specified time; otherwise returns a PreconditionFailed.<br>Type: dateTime<br>Default: None   | No       |
| <i>CopySourceIfModifiedSince</i>   | Copies the object if it has been modified since the specified time; otherwise returns an error.<br>Type: dateTime<br>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

| Name                | Description  |
|---------------------|--|
| <i>Etag</i>         | Returns the etag of the new object. The ETag only reflects changes to the contents of an object, not its metadata.<br>Type: String<br>Ancestor: CopyObjectResult |
| <i>LastModified</i> | Returns the date the object was last modified.<br>Type: String<br>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. 3).

## 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>1D9FVRAYCP1VJEXAMPLE=</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>"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>1D9FVRAYCP1VJEXAMPLE=</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>"828ef3fdfa96f00ad9f27c383fc9ac7f"</ETag>
    <LastModified>2008-02-18T13:54:10.183Z</LastModified>
  </CopyObjectResponse>
</CopyObjectResponse>
```

## Related Resources

- [Copying Objects](#)
- [PutObject](#) (p. 186)
- [PutObjectInline](#) (p. 184)

## GetObject

The `GetObject` operation returns the latest version of an object. If you try to `GetObject` an object that has a Delete Marker as its latest 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](#) (p. 198) 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>1D9FVRAYCP1VJEXAMPLE=</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>"828ef3fdfa96f00ad9f27c383fc9ac7f"</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 \(p. 195\)](#)
- 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 \(p. 198\)](#)

## 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>1D9FVRAYCP1VJEXAMPLE=</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>bca1fffd86f41161ca5fb16fd081034f</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>1D9FVRAYCP1VJEXAMPLE=</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>"828ef3fd9a96f00ad9f27c383fc9ac7f"</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>"828ef3fd9a96f00ad9f27c383fc9ac7f"</IfMatch>
<AWSAccessKeyId>1D9FVRAYCP1VJEXAMPLE=</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>1D9FVRAYCP1VJEXAMPLE=</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>"828ef3fdfa96f00ad9f27c383fc9ac7f"</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](#) (p. 198).

## Related Resources

[Operations on Objects](#) (p. 183)

# GetObjectExtended

`GetObjectExtended` is exactly like [GetObject](#) (p. 193), 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

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> 1D9FVRAYCP1VJEXAMPLE=</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

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>1D9FVRAYCP1VJEXAMPLE=</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, see [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

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>1D9FVRAYCP1VJEXAMPLE=</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.

# 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 S3 Getting Started Guide</a> | The Getting Started Guide provides a quick tutorial of the service based on a simple use case. Examples and instructions for Java, Perl, PHP, C#, Python, and Ruby are included.   |
| <a href="#">Amazon S3 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="#">AWS Developer Resource Center</a>   | A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.                             |
| <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.  |

| Resource                          | Description  |
|-----------------------------------|--|
| <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

---

This document history is associated with the 2006-03-01 release of Amazon S3. This guide was last updated on 17 October 2011.

The following table describes the important changes since the last release of the *Amazon S3 API Reference*.

| Change   | Description   | Release Date     |
|--|---|------------------|
| 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 this release. |
| 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. 142)</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, <code>Upload Part (Copy)</code> . 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> . To learn more about the new API, see <a href="#">Upload Part - Copy (p. 156)</a> . | 21 June 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.   | 6 June 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</a> (p. 99)</li> <li>• <a href="#">GET Bucket website</a> (p. 62)</li> <li>• <a href="#">DELETE Bucket website</a> (p. 23)</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> | 17 February 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</a> (p. 106).   | 14 January 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> . For multipart upload API information, see <a href="#">Initiate Multipart Upload</a> (p. 149), <a href="#">Upload Part</a> (p. 153), <a href="#">Complete Multipart Upload</a> (p. 161), <a href="#">List Parts</a> (p. 168), and <a href="#">List Multipart Uploads</a> (p. 65)   | 9 December 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> . For multipart upload API information, see <a href="#">Initiate Multipart Upload</a> (p. 149), <a href="#">Upload Part</a> (p. 153), <a href="#">Complete Multipart Upload</a> (p. 161), <a href="#">List Parts</a> (p. 168), and <a href="#">List Multipart Uploads</a> (p. 65)  | 10 November 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</a> (p. 42) and <a href="#">PUT Bucket notification</a> (p. 42).   | 14 July 2010     |
| 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.  | 6 July 2010      |

| Change                          | Description   | Release Date      |
|---------------------------------|---|-------------------|
| 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. 129)</a> .   | 12 May 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. 37)</a> and <a href="#">PUT Bucket (p. 74)</a> .   | 28 April 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. 106)</a> , <a href="#">DELETE Object (p. 102)</a> , <a href="#">PUT Object (p. 129)</a> , <a href="#">PUT Object Copy (p. 142)</a> , or <a href="#">POST Object (p. 122)</a> . The SOAP API does not support versioned objects. | 8 February 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> .  | 2 December 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 APIs 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.  | 11 November 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="http://developer.amazonwebservices.com/connect/entry.jspa?externalID=123&amp;categoryID=48">http://developer.amazonwebservices.com/connect/entry.jspa?externalID=123&amp;categoryID=48</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.  | 16 September 2009 |

# Glossary

---

|                      |  |
|----------------------|--|
| 100-continue         | A method that enables a client to see if a server can accept a request before actually sending it. For large <code>PUT</code> s, 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 <code>photos/puppy.jpg</code> is stored in the <code>johnsmith</code> bucket, then it is addressable using the URL<br><code>http://johnsmith.s3.amazonaws.com/photos/puppy.jpg</code>   |
| canned access policy | A standard access control policy that you can apply to a bucket or object. Valid Values: <code>private</code>   <code>public-read</code>   <code>public-read-write</code>   <code>authenticated-read</code>   <code>bucket-owner-read</code>   <code>bucket-owner-full-control</code>  |
| 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 <code>http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl</code> , 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 <code>Content-Type</code> . 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> |
|------------------|---|

# Index

## A

abort upload, 166  
API  
    REST, 11

## B

buckets  
    creating, 175  
    putting, 23, 62, 74, 78, 82, 84, 95, 99  
    REST operations, 18  
    SOAP operations, 174

## C

changes, 204  
common response headers, 14  
complete upload, 161  
content-MD5, 122, 129, 135, 186  
Content-MD5, 184  
COPY, REST, 142  
copying objects  
    REST, 142  
    SOAP, 189  
CopyObject  
    errors, 192  
CopyObject, SOAP, 189  
CreateBucket  
    SOAP, 175  
creating buckets  
    SOAP, 175

## D

data corruption  
    preventing, 122, 129, 135, 184, 186  
DELETE  
    REST, 19, 21, 102  
DeleteBucket, 176  
deleting objects  
    SOAP, 198  
duplicating objects, 189

## E

errors  
    CopyObject, 192  
    list, 3

## G

GET  
    policy, REST, 15  
    REST, 25, 42, 45, 57, 59, 112  
GET, REST, 32, 35, 37, 39, 106, 116  
GetBucketAccessControlPolicy

    SOAP, 179  
GetBucketLoggingStatus  
    SOAP, 181  
GetObject  
    SOAP, 193  
GetObjectAccessControlPolicy  
    SOAP, 199  
GetObjectExtended  
    SOAP, 198  
guide organization, 1

## H

HEAD object  
    REST, 118  
headers, request  
    REST, 12

## I

initiate, 149

## L

large object upload  
    abort, 166  
    complete, 161  
    initiate, 149  
    list, 65  
    list parts, 168  
    upload part, 153, 156  
list multipart uploads, 65  
list parts, 168  
ListAllMyBuckets  
    SOAP, 173  
ListBucket  
    SOAP, 176

## N

notification, 42, 89

## O

objects  
    copying, 142, 189  
    REST operations, 101  
    SOAP operations, 183  
organization of guide, 1

## P

POST, REST, 122  
PUT  
    REST, 23, 62, 74, 78, 82, 84, 89, 93, 95, 99  
PUT, REST, 129, 135  
PutObject  
    SOAP, 186  
PutObjectInline  
    SOAP, 184  
putting buckets

REST, 23, 62, 74, 78, 82, 84, 95, 99  
putting Request Payments  
REST, 93

## R

request headers, 12  
requestPayments  
  putting, 93  
resources, related, 202  
response headers, common, 14  
REST  
  API, 11  
  bucket operations, 18  
  COPY, 142  
  DELETE, 19, 21, 102  
  GET, 15, 25, 42, 45, 57, 59, 112  
  GET object, 32, 35, 37, 39, 106, 116, 153, 156  
  HEAD object, 118  
  object operations, 101  
  POST object, 122  
  PUT, 23, 62, 74, 78, 82, 84, 89, 93, 95, 99  
  PUT object, 129, 135  
  service operations, 15

## S

service  
  REST operations, 15  
  SOAP operations, 173  
SetBucketAccessControlPolicy  
  SOAP, 180  
SetBucketLoggingStatus  
  SOAP, 182  
SetObjectAccessControlPolicy  
  SOAP, 200  
SOAP  
  bucket operations, 174  
  CopyObject, 189  
  CreateBucket, 175  
  DeleteBucket, 176  
  DeleteObject, 198  
  GetBucketAccessControlPolicy, 179  
  GetBucketLoggingStatus, 181  
  GetObject, 193  
  GetObjectAccessControlPolicy, 199  
  GetObjectExtended, 198  
  ListAllMyBuckets, 173  
  ListBucket, 176  
  object operations, 183  
  PutObject, 186  
  PutObjectInline, 184  
  service operations, 173  
  SetBucketAccessControlPolicy, 180  
  SetBucketLoggingStatus, 182  
  SetObjectAccessControlPolicy, 200

## U

upload part, 153, 156

## W

WSDL, 2