Welcome to Amazon S3

Topics

- Who Should Read this Guide
- How to Give Us Feedback
- How This Guide Is Organized
- Amazon S3 Resources

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 (EC2).

This section describes who should read this guide, how the guide is organized, and other resources related to Amazon S3.

We hope you find the service to be easy-to-use, reliable, and inexpensive. If you want to provide feedback to the Amazon S3 development team, please post a message to the Amazon S3 Discussion Forum or the Feedback link at the top of every page in the HTML version of this guide.

Who Should Read this Guide

This guide is for developers who are creating libraries to implement the Amazon S3 API. This audience can use the API reference to learn how the HTTP packets should look for particular requests.

Required Knowledge and Skills

Use of this guide assumes you are familiar with the following:

- XML (go to <u>W3 Schools XML Tutorial</u>)
- Basic understanding of web services (go to <u>W3</u>
 <u>Schools Web Services Tutorial</u>))
- A programming language for consuming a web service and any related tools

You should also have read the *Amazon S3 Getting Started Guide*. For more information, go to <u>Amazon S3 Getting Started Guide</u>.

How to Give Us Feedback

The online version of this guide provides a link at the top of each page that enables you to enter feedback about this guide. We strive to make our guides as complete, error free, and easy to read as possible. You can help by giving us feedback. Thank you in advance!



How This Guide Is Organized

This guide is organized into several major sections described in the following table.

Information	Relevant Sections
Common request headers	Common Request Headers describes parameters that can or, in some cases, must be used in each request.
Common response headers	Common Response Headers describes parameters that can or, in some cases, must be used in each request.
Error codes	List of Error Codes lists Amazon S3 error codes and descriptions.
REST API	REST API describes the REST API in detail.
SOAP API	SOAP API describes the SOAP API in detail.
Typographic and symbol conventions	Document Conventions explains the typographic conventions used in this guide.

Each section is written to stand on its own, so you should be able to look up the information you need and go back to work. However, you can also read through the major sections sequentially to get in-depth knowledge about the Amazon S3.

Amazon S3 Resources

Following is a table that lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon S3 Getting Started Guide	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.
Amazon S3 Developer Guide	The developer guide describes how to accomplish tasks using Amazon S3 operations.
Amazon S3Technical FAQ	The FAQ covers the top 20 questions developers have asked about this product.
Amazon S3 Release Notes	The Release Notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
AWS Management Console	The console allows you to perform most of the functions of Amazon S3without programming.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.
AWS Premium Support	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.
Amazon S3 product information	The primary web page for information about Amazon S3.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

What's New

This What's New is associated with the 2006-03-01 release of Amazon S3. This guide was last updated on February 08, 2010.

The following table describes the important changes since the last release of the Amazon S3 Developer Guide.

Change	Description	Release Date
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 GET Object , DELETE Object , PUT Object Copy , or POST Object . 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 s3-us-west-1.amazonaws.com. For more information, see How to Select a Region for Your Buckets .	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, http://developer.amazonwebservices.com/connect/entry.jspa? externalID=123&categoryID=48 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

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 Contact Us .	403 Forbidden	Client
Ambiguous Grant By Email Address	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
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	Client

		Request	
Illegal Version ing Configuration Exception	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
InternalError	We encountered an internal error. Please try again.	500 Internal Server Error	Server
InvalidAccessKeyId	The AWS Access Key Id you provided does not exist in our records.	403 Forbidden	Client
InvalidAddressingHeader	You must specify the Anonymous role.	N/A	Client
InvalidArgument	Invalid Argument	400 Bad Request	Client
InvalidBucketName	The specified bucket is not valid.	400 Bad Request	Client
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 <u>How to Select a Region for Your Buckets</u> .	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
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
MalformedACLError	The XML you provided was not well-formed or did not validate against our published schema.	400 Bad Request	Client
MalformedPOSTRequest	The body of your POST request is not well-formed multipart/form-data.	400 Bad Request	Client
MalformedXML	This happens when the user sends 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
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: http://aws.amazon.com/s3	403 Forbidden	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 pre-conditions 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 REST Authentication and SOAP Authentication for details.	403 Forbidden	Client
SlowDown	Please reduce your request rate.	503 Service Unavailable	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 or the response also contains information about the error. The following sample error response shows the structure of response elements common to all REST error responses.

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

The following table explains the REST error response elements

Name	Description
Code	The error code is a string that uniquely identifies an error condition. It is meant to be read and understood by programs that detect and handle errors by type. For more information, see <u>List of Error Codes</u> .
	Type: String

	Ancestor: Error
Error	Container for all error elements.
	Type: Container
	Ancestor: None
Message	The error message contains a generic description of the error condition in English. It is intended for a human audience. Simple programs display the message directly to the end user if they encounter an error condition they don't know how or don't care to handle. Sophisticated programs with more exhaustive error handling and proper internationalization are more likely to ignore the error message.
	Type: String
	Ancestor: Error
RequestId	ID of the request associated with the error.
	Type: String
	Ancestor: Error
Resource	The bucket or object that is involved in the error.
	Type: String
	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, see <u>Using REST Error Response Headers</u>.

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.</Faultcode>
    <Detail>
        <Key>Fred</Key>
```

</Detail>
</soapenv:Fault>
</soapenv:Body>

The following table explains the SOAP error response elements

Name	Description
Detail	Container for the key involved in the error
	Type: Container
	Ancestor: Body.Fault
Fault	Container for error information.
	Type: Container
	Ancestor: Body
Faultcode	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 <u>List of Error Codes</u> .
	Type: String
	Ancestor: Body.Fault
Faultstring	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.
	Type: String
	Ancestor: Body.Fault
Key	Identifies the key involved in the error
	Type: String
	Ancestor: Body.Fault

REST API

Topics

- Common Request Headers
- Common Response Headers
- Operations on the Service
- Operations on Buckets
- Operations on Objects

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

The examples in this guide use the newer virtual hostedstyle method for accessing buckets instead of the pathstyle. Although the path-style is still supported for legacy applications, we recommend using the virtual-hosted style where applicable. For more information, see <u>Working with Amazon S3 Buckets</u>

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 0PN5J17HBGZHT7JJ3X82:k3nL7gH3+Padh1
```

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 0PN5J17HBGZHT7JJ3X82:k3nL7gH3+Padh1
```

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.	Yes
	Type: String	
	Default: None	
Content- Length	Length of the message (without the headers) according to RFC 2616.	Conditional
	Type: String	
	Default: None	
	Condition: Required for PUTs and operations that load XML, such as logging and ACLs.	
Content-Type	The content type of the resource. Example: text/plain	No
	Type: String	
	Default: None	
Date	The current date and time according to the requester. Example: Wed, 01 Mar 2009 12:00:00 GMT	Yes
	Type: String	
	Default: None	
Host	For path-style requests, the value is s3.amazonaws.com. For virtual-style requests, the value is BucketName.s3.amazonaws.com. For more information, go to Virtual Hosting in the Amazon Simple Storage Service Developer Guide.	Conditional

	Type: String	
	Default: None	
	Condition: Required for HTTP 1.1 (most toolkits add this header automatically); optional for HTTP/1.0 requests.	
x-amz- security- token	The security tokens for operations that use Amazon DevPay. Each request that uses Amazon DevPay requires two x-amz-security-token headers: one for the product token and one for the user token.	Conditional
	When Amazon S3 receives an authenticated request, it compares the computed signature with the provided signature. Improperly formatted multivalue headers used to calculate a signature can cause authentication issues.	
	Type: String	
	Default: None	
	Condition: Required for requests that use Amazon DevPay.	

Common Response Headers

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

Name	Description
Content-	The length in bytes of the body in the response.
Length	Type: String
	Default: None
Connection	Specfies whether the connection to the server is open or closed.
	Type: Enum
	Valid Values: open close
	Default: None
Date	The date and time Amazon S3 responded, for example, Wed, 01 Mar 2009 12:00:00 GMT.
	Type: String
	Default: None
ETag	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. Amazon S3 checks the object against the provided MD5 value. If they do not match, Amazon S3 returns an error. The ETag only reflects changes to the contents of an object, not its metadata.
	Type: String
Server	The name of the server that created the response.
	Type: String
	Default: None
x-amz-	Specifies whether the object returned was (true) or was not (false) a Delete Marker.
delete- marker	Type: Boolean
	Valid Values: true false

	Default: false	
x-amz-id-2	A special token that helps AWS troubleshoot problems. Type: String	
	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.	
	Type: String	
	Default: None	
x-amz- version-id	The version of the object. When you enable versioning, Amazon S3 generates a random number for objects added to a bucket. The value is UTF-8 encoded and URL ready. When you PUT an object in a bucket where versioning has been suspended, the version ID is always null.	
	Type: String	
	Valid Values: null any	
	URL-ready, UTF-8 encoded string	
	Default: null	

Operations on the Service

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.

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.

Response Elements

Name	Description
Bucket	Container for bucket information.
	Type: Container
	Children: Name, CreationDate
	Ancestor: ListAllMyBucketsResult.Buckets
Buckets	Container for one or more buckets.
	Type: Container
	Children: Bucket
	Ancestor: ListAllMyBucketsResult
CreationDate	Date the bucket was created.
	Type: date (of the form yyyy-mm-ddThh:mm:ss.timezone, e.g., 2009-02-03T16:45:09.000Z)
	Ancestor: ListAllMyBucketsResult.Buckets.Bucket
DisplayName	Bucket owner's display name.
	Type: String
	Ancestor: ListAllMyBucketsResult.Owner
ID	Bucket owner's user ID.
	Type: String

	Ancestor: ListAllMyBucketsResult.Owner
ListAllMyBucketsResult	Container for response.
	Type: Container
	Children: Owner, Buckets
	Ancestor: None
Name	Bucket's name.
	Type: String
	Ancestor: ListAllMyBucketsResult.Buckets.Bucket
Owner	Container for bucket owner information.
	Type: Container
	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.

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:xQE0diMbLRepd1
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://doc.s3.amazonaws</pre>
  <0wner>
    <ID>bcaf1ffd86f461ca5fb16fd081034f</ID>
    <DisplayName>webfile/DisplayName>
  </0wner>
  <Buckets>
    <Bucket>
      <Name>quotes;/Name>
      <CreationDate>2006-02-03T16:45:09.000Z</Creation</pre>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <CreationDate>2006-02-03T16:41:58.000Z</Creation</pre>
    </Bucket>
  </Buckets>
```

</ListAllMyBucketsResult>

Related Resources

- GET Bucket (List Objects)
- GET Object

Operations on Buckets

Topics

•

DELETE Bucket

- GET Bucket (List Objects)
- GET Bucket acl
- GET Bucket location
- GET Bucket logging
- GET Bucket Object versions
- GET Bucket requestPayment
- GET Bucket versioning
- PUT Bucket
- PUT Bucket acl
- PUT Bucket logging
- PUT Bucket requestPayment
- PUT Bucket versioning

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



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. Only the owner of a bucket can delete it, regardless of the bucket's access control policy.

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.

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.

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.

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:xQE0diMbLRepd1
```

Sample Response

```
HTTP/1.1 204 No Content
x-amz-id-2: JuKZqmXuiwFeDQxhD7M8KtsKobSzWA1QEjLbTMTagk
x-amz-request-id: 32FE2CEB32F5EE25
Date: Wed, 01 Mar 2009 12:00:00 GMT
Connection: close
Server: AmazonS3
```

Related Resources

- PUT Bucket
- DELETE Object

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.

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
delimiter	A delimiter is a character you use to group keys. All keys that contain the same string between the prefix and the first occurrence of the delimiter are grouped under a single result element, <i>CommonPrefixes</i> . These keys are not returned elsewhere in the response.	No
	Type: String	
	Default: None	
key- marker	Specifies the key to start with when listing objects in a bucket. Amazon S3 lists objects in alphabetical order.	No
	Type: String	
	Default: None	
max-keys	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 <code>max-keys</code> was exceeded, the response contains <code>sisTruncated>true</code> . To return the additional keys, see <code>key-marker</code> .	No

	Type: String	
	Default: 1000	
prefix	Limits the response to keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using <i>prefix</i> to make groups in the same way you'd use a folder in a file system.)	No
	Type: String	
	Default: None	

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.

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.

Response Elements

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

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

Prefix	Keys that begin with the indicated prefix.
	Type: String
	Ancestor: ListBucketResult
Size	Size in bytes of the object.
	Type: String
	Ancestor: ListBucketResult.Contents
StorageClass	Always STANDARD.
	Type: String
	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.

Examples

Sample Request

This requests 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:xQE0diMbLRepd1
Content-Type: text/plain
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2</pre>
    <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>2009-10-12T17:50:30.000Z
        <ETag>&quot; fba9dede5f27731c9771645a39863328&c
         <Size>434234</Size>
        <StorageClass>STANDARD</StorageClass>
         <0wner>
             <ID>8a6925ce4a7f21c32aa379004fef</ID>
             <DisplayName>mtd@amazon.com</DisplayName>
        </0wner>
```

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:xQE0diMbLRepd1
```

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxexeEHnc
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/2</pre>
  <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/LastModifi
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;
    <Size>5</Size>
    <StorageClass>STANDARD</StorageClass>
    <0wner>
      <ID>bcaf161ca5fb16fd081034f</ID>
      <DisplayName>webfile/DisplayName>
     </0wner>
  </Contents>
  <Contents>
    <Key>Neo</Key>
    <LastModified>2006-01-01T12:00:00.000Z/LastModifi
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;
    <Size>4</Size>
    <StorageClass>STANDARD</StorageClass>
     <0wner>
      <ID>bcaf1ffd86a5fb16fd081034f</ID>
      <DisplayName>webfile/DisplayName>
    </0wner>
 </Contents>
</ListBucketResult>
```

Sample Request Using Prefix and Delimiter

This example lists rolls up all objects for each URI that starts with photos/2006/, has some path element, and ends with the first occurrence of the delimiter "/", for example, each of the following would become a CommonPrefixes entry: photos/2006/January/, photos/2006/February/, and photos/2006/March/

```
GET ?prefix=photos/2006/&delimiter=/ HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepd1
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2</pre>
  <Name>johnsmith</Name>
  <Prefix>photos/2006/</Prefix>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</isTruncated>
  <Contents>
    <Key>photos/2006/index.html</Key>
    <LastModified>2009-01-01T12:00:00.000Z/LastModifi
    <ETag>"celacdafcc879d7eee54cf4e97334078"</ETag>
    <Size>1234</Size>
    <0wner>
      <ID>214153b66967d86f031c7487b4566cb1b</ID>
      <DisplayName>John Smith/DisplayName>
    </0wner>
```

Related Resources

- GET Object
- PUT Object
- PUT Bucket

GET Bucket acl

Description

This implementation of the GET operation uses the *ac1* sub-resource 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.

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.

Response Elements

Name	Description
AccessControlList	Container for ACL information.
	Type: Container
	Ancestry: AccessControlPolicy
AccessControlPolicy	Container for the reponse.
	Type: Container
	Ancestry: None
DisplayName	Bucket owner's display name. This is returned only if the owner's e-mail address (or the forum name, if configured) can be determined from the <i>ID</i> .
	Type: String
	Ancestry: AccessControlPolicy.Owner
Grant	Container for Grantee and Permission.
	Type: Container
	Ancestry: AccessControlPolicy.AccessControlList
Grantee	Container for <code>DisplayName</code> and <code>ID</code> of the person being granted permissions.
	Type: Container
	Ancestry: AccessControlPolicy.AccessControlList.Grant
ID	Bucket owner's ID.

	Type: String
	Ancestry: AccessControlPolicy.Owner
Owner	Container for bucket owner information.
	Type: Container
	Ancestry: AccessControlPolicy
Permission	Permission given to the <i>Grantee</i> for bucket.
	Type: String
	Valid Values: FULL_CONTROL WRITE WRITE_ACP READ READ_ACP
	Ancestry: AccessControlPolicy.AccessControlList.Grant

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses.

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/cRonhpaE
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9
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>
  <0wner>
    <ID>8a6925ce4adee97f21c32aa379004fef</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName</pre>
  </0wner>
  <AccessControlList>
    <Grant>
```

Related Resources

• GET Bucket Objects

GET Bucket location

Description

This implementation of the GET operation uses the *location* sub-resource to return a bucket's Region. You set the bucket's Region using the *LocationContraint* request parameter in a PUT *Bucket* request. For more information, see

PUT Bucket.

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 <u>Common Request Headers</u>.

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 <u>Common Response Headers</u>.

Response Elements

Name	Description
LocationConstraint	Specifies the Region where the bucket resides.
	Type: String
	Valid Values: EU us-west-1 empty string (for the US Classic Region)
	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</pre>
```

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.

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:JUtd9kkJFjbKbk
```

Sample Response

```
<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint xmlns="http://s3.amazonaws.com/doc</pre>
```

Related Resources

- GET Bucket Objects
- PUT Bucket

GET Bucket logging



Description

This implementation of the GET operation uses the *logging* sub-resource 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.

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.

Response Elements

Name	Description
BucketLoggingStatus	Container for the response.
	Type: Container
	Ancestry: None
EmailAddress	E-mail address of the person whose logging permissions are displayed.
	Type: String
	Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant.Grantee
Grant	Container for Grantee and Permission.
	Type: Container
	Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants
Grantee	Container for <i>EmailAddress</i> of the person whose logging permissions are displayed.
	Type: Container
	Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant
LoggingEnabled	Container for logging information. This element and its children are present when logging is enabled, otherwise, this element and its children are absent.
	Type: Container
	Ancestry: BucketLoggingStatus
Permission	Logging permissions assigned to the <i>Grantee</i> for the bucket.

	Type: String
	Valid Values: FULL_CONTROL READ WRITE
	Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant
TargetBucket	Specifies the bucket whose logging status is being returned. This element specifies the bucket where server access logs will be delivered.
	Type: String
	Ancestry: BucketLoggingStatus.LoggingEnabled
TargetGrants	Container for granting information.
	Type: Container
	Ancestry: BucketLoggingStatus.LoggingEnabled
TargetPrefix	Specifies the prefix for the keys that the log files are being stored under.
	Type: String
	Ancestry: BucketLoggingStatus.LoggingEnabled

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses.

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/cRonhpaE
```

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.cc</pre>
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XML</pre>
          xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress</pre>
        </Grantee>
        <Permission>READ</Permission>
```

```
</Grant>
  </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Sample Response Showing a Disabled Logging Status

```
HTTP/1.1 200 0K
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.co
```

Related Resources

- PUT Bucket
- PUT Bucket logging

GET Bucket Object versions

Description

You can use the *versions* sub-resource 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.

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
delimiter	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> . Type: String Default: None	No
key- marker	Specifies the key in the bucket that you want to start listing from. Also, see <pre>version-id-marker.</pre> Type: String Default: None	No
max-keys	Sets the maximum number of keys returned in the response body. The response might contain fewer keys, but will never contain more. If additional keys satisfy the search criteria, but were not returned because <code>max-keys</code> was exceeded, the response contains <code><istruncated>true</istruncated></code> . To return the additional keys, see <code>key-marker</code> and <code>version-id-marker</code> .	No

	Type: String Default: 1000	
prefix	Use this parameter to select only those keys that begin with the specified prefix. You can use prefixes to separate a bucket into different groupings of keys. (You can think of using <code>prefix</code> to make groups in the same way you'd use a folder in a file system.) You can use <code>prefix</code> with <code>delimiter</code> to roll up numerous objects into a single result under <code>commonPrefixes</code> . Also, see <code>delimiter</code> . Type: String	No
	Default: None	
version-id- marker	Specifies the object version you want to start listing from. Also, see <i>key-marker</i> .	No
marker	Type: String	
	Default: None	
	Valid Values: Valid version ID Default	
	Constraint: May not be an empty string	

Request Headers

This implementation of the operation only uses request headers common to all operations. For more information, see Common Request Headers.

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.

Response Elements

Name	Description
DeleteMarker	Container for an object that is a Delete Marker.
	Type: Container
	Children: Key, VersionId, IsLatest, LastModified, Owner
	Ancestor: ListVersionsResult
DisplayName	Object owner's name.
	Type: String
	Ancestor: ListVersionsResult.Version.Owner ListVersionsResult.DeleteMarker.Owner
ЕТад	The entity tag is an MD5 hash of the object. The ETag only reflects changes to the contents of an object, not its metadata.
	Type: String
	Ancestor: ListVersionsResult.Version
ID	Object owner's ID.
	Type: Boolean
	Ancestor: ListVersionsResult.Version.Owner ListVersionsResult.DeleteMarker.Owner
IsLatest	Specifies whether the object is (true) or is not (false) the latest version of an object.

	Type: Boolean
	Valid Values: true false
	Ancestor: ListVersionsResult.Version ListVersionsResult.DeleteMarker
IsTruncated	A flag that indicates whether (true) or not (false) Amazon S3 returned all of the results that satisfied the search criteria. If your results were truncated, you can make a follow-up paginated request using the <code>NextKeyMarker</code> and <code>NextVersionIdMarker</code> response parameters as a starting place in another request to return the rest of the results.
	Type: Boolean
	Valid Values: true false
	Ancestor: ListVersionsResult
Key	The object's key.
	Type: String
	Ancestor: ListVersionsResult.Version ListVersionsResult.DeleteMarker
KeyMarker	Marks the last <i>Key</i> returned in a truncated response.
	Type: String
	Ancestor: ListVersionsResult
LastModified	Date and time the object was last modified.
	Type: Date
	Ancestor: ListVersionsResult.Version ListVersionsResult.DeleteMarker
ListVersionsResult	Container for the result.
	Type: Container
	Children: All elements in the response
	Ancestor: ListVersionsResult
MaxKeys	Specifies the maximum number of objects to return.
	Type: String
	Default: 1000
	Valid Values: Integers from 1 to 1000, inclusive
	Ancestor: ListVersionsResult
Name	Bucket owner's name.
	Type: String

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

	Type: String
	Ancestor: ListVersionsResult

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses.

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/cRonhpaE
```

Sample Response to GET Versions

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc</pre>
    <Name>bucket</Name>
    <Prefix>my</Prefix>
    <KeyMarker/>
    <VersionIdMarker/>
    <MaxKeys>5</MaxKeys>
    <IsTruncated>false</IsTruncated>
    <Version>
        <Key>my-image.jpg</Key>
        <VersionId>3/L4kqtJl40Nr8X8qdRQBpUMLUo</VersionId>3/L4kqtJl40Nr8X8qdRQBpUMLUo
        <IsLatest>true</IsLatest>
         <LastModified>2009-10-12T17:50:30.000Z
        <ETag>&quot; fba9dede5f27731c9771645a39863328&c
        <Size>434234</Size>
        <StorageClass>STANDARD</StorageClass>
```

```
<0wner>
         <ID>8a6925ce4adf58897f21c32aa379004fef</II
        <DisplayName>mtd@amazon.com</DisplayName>
    </0wner>
</Version>
<DeleteMarker>
    <Key>my-second-image.jpg</Key>
    <VersionId>03jpff543dhffds434rfdsFDN943fdsFkdn
    <IsLatest>true</IsLatest>
    <LastModified>2009-11-12T17:50:30.000Z/LastModified>2009-11-12T17:50:30.000Z
    <0wner>
        <ID>8a6925ce4adf7f21c32aa379004fef</ID>
        <DisplayName>mtd@amazon.com</DisplayName>
    </0wner>
</DeleteMarker>
<Version>
    <Key>my-second-image.jpg</Key>
    <VersionId>QUpfdndhfd8438MNFDN93jdnJFkdmgnh893
    <IsLatest>false</IsLatest>
    <LastModified>2009-10-10T17:50:30.000Z/LastModified>2009-10-10T17:50:30.000Z
    <ETag>&quot;9b2cf535f27731c974343645a3985328&c
    <Size>166434</Size>
    <StorageClass>STANDARD</StorageClass>
    <0wner>
         <ID>8a6925ce421c32aa379004fef</ID>
        <DisplayName>mtd@amazon.com</DisplayName>
    </0wner>
</Version>
<DeleteMarker>
    <Key>my-third-image.jpg</Key>
    <VersionId>03jpff543dhffds434rfdsFDN943fdsFkdn
    <IsLatest>true</IsLatest>
    <LastModified>2009-10-15T17:50:30.000Z/LastModified>2009-10-15T17:50:30.000Z
    <0wner>
        <ID>8a6925ce4adf532aa379004fef</ID>
        <DisplayName>mtd@amazon.com</DisplayName>
    </0wner>
```

```
<
```

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) libcur
Host: s3.integ.amazon.com
Pragma: no-cache
Accept: image/gif, image/x-xbitmap, image/jpeg, image/
Date: Thu, 10 Dec 2009 22:46:32 +0000
Authorization: AWS 0A1N5HNJ7ZX40NG0KT02:U1j5vNnJfzmiv3
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc</pre>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key2</KeyMarker>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>I5VhmK6CDDdQ5Pwfe1gcHZWmHDpcv7gfmfc29UE
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-09T00:19:04.000Z/LastModifi
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;
    <Size>217</Size>
    <0wner>
      <ID>0fa3c2dd7e785a220ab556f8a401610b750c0b6d62e1
    </0wner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
  <DeleteMarker>
    <Key>sourcekey</Key>
    <VersionId>qDhprLU80sAlCFLu2DWgXAEDgKzWarn-HS JU01
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-10T16:38:11.000Z/LastModifi
    <0wner>
      <ID>0fa3c2dd7e785a2d1667a1e068364</ID>
    </0wner>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxxQ7ezLaL5JN2Sislq66Syxxo0k7uHTUpb9qii
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z/LastModifi
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;
    <Size>217</Size>
    <0wner>
```

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/cRonhpaE
```

```
<ID>0fa3c2dd7e78dd1667a1e068364</ID>
    </0wner>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxxQ7ezLaL5JN2Sislg66Syxxo0k7uHTUpb9gij
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z/LastModifi
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;
    <Size>217</Size>
    <0wner>
      <ID>0fa3c2d87663dd1667a1e068364</ID>
    </0wner>
    <StorageClass>STANDARD</StorageClass>
  </Version>
</ListVersionsResult>
```

Sample Request Using key-marker and version-idmarker 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=t46Ze
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaE
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc</pre>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>t46ZenlYTZBnj</versionIdMarker>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <DeleteMarker>
    <Key>sourcekey</Key>
    <VersionId>qDhprLU80sAlCFLu2DWgXAEDgKzWarn-HS JU01
    <IsLatest>true</IsLatest>
    <LastModified>2009-12-10T16:38:11.000Z/LastModifi
    <0wner>
      <ID>0fa3c2dd7e785a2201667a1e068364</ID>
    </0wner>
  </DeleteMarker>
  <Version>
    <Key>sourcekey</Key>
    <VersionId>wxxQ7ezLaL5JN2Sislq66Syxxo0k7uHTUpb9qij
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-10T16:37:44.000Z/LastModifi
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;
    <Size>217</Size>
    <0wner>
      <ID>0fa3c2dd7e785a663dd1667a1e068364</ID>
    </0wner>
    <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=t46Z@Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaE
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://s3.amazonaws.com/doc</pre>
  <Name>mtp-versioning-fresh</Name>
  <Prefix/>
  <KeyMarker>key3</KeyMarker>
  <VersionIdMarker>null</versionIdMarker>
  <NextKeyMarker>key3</NextKeyMarker>
  <NextVersionIdMarker>d-d309mfjFrUmoQ0DBsVqmcMV150I.
  <MaxKeys>2</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <Version>
    <Key>key3</Key>
    <VersionId>8XECiENpj8pydEDJdd- VRrvaGKAHOaGMNW7tg6
    <IsLatest>false</IsLatest>
    <LastModified>2009-12-09T00:18:23.000Z/LastModifi
    <ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;
    <Size>217</Size>
    <0wner>
      <ID>0fa3c2dd7e785a220ab667a1e068364</ID>
    </0wner>
    <StorageClass>STANDARD</StorageClass>
```

```
</Pre>
<Key>key3/Key>
<VersionId>d-d309mfjFri40QYukDozqBt3UmoQ0DBsVqmcM\
<IsLatest>false</IsLatest>
<LastModified>2009-12-09T00:18:08.000Z/LastModifi
<ETag>&quot;396fefef536d5ce46c7537ecf978a360&quot;
<Size>217</Size>
<0wner>
<ID>0fa3c2dd7e785a220ad1667a1e068364</ID>
</owner>
<StorageClass>STANDARD/StorageClass>
</Pre>

<p
```

Sample Request Using prefix and delimiter

The following request groups under CommonPrefixes object versions whose name starts with the value of prefix (photos) and ends with the value for the delimiter (/), in this example. The resposne includes photos/January/, photos/February, and photos/March.

```
GET /?versions&prefix=photos&delimiter=/ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaE
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2</pre>
  <Name>johnsmith</Name>
  Prefix>photos/</Prefix>
 Marker/>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</isTruncated>
  <Contents>
    <Key>photos/index.html</Key>
    <LastModified>2009-01-01T12:00:00.000Z/LastModifi
    <ETag>"celacdafcc879d7eee54cf4e97334078"</ETag>
    <Size>1234</Size>
    <0wner>
      <ID>214153b66967d86f031c7249d1d9a80249109428335c
      ID>
      <DisplayName>John Smith
    </0wner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/January/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Related Resources

- GET Bucket Objects
- GET Object
- PUT Object
- DELETE Object

GET Bucket requestPayment

Description

This implementation of the GET operation uses the *requestPayment* sub-resource 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 <u>Common Request Headers</u>.

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.

Response Elements

Name	Description
Payer	Specifies who pays for the download and request fees.
	Type: Enum
	Valid Values: Requester BucketOwner
	Ancestor: RequestPaymentConfiguration
RequestPaymentConfiguration	Container for <i>Payer</i> .
	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.

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:xQE0diMbLRepd1
```

Sample Response

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)

GET Bucket versioning

Description

This implementation of the GET operation uses the *versioning* sub-resource 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:

• If you suspended versioning on a bucket, the response is:

• If you never enabled (or suspended) versioning on a bucket, the response is:

<VersioningConfiguration xmlns="http://s3.amazonaw</pre>

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.

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.

Response Elements

This implementation of GET returns the following response elements.

Name	Description
MfaDelete	Specifies whether MFA delete is enabled in the bucket versioning configuration
	Type: Enum
	Valid Values: Disabled Enabled
	Ancestor: VersioningConfiguration
Status	The versioning state of the bucket.
	Type: Enum
	Valid Values: Suspended Enabled
	Ancestor: VersioningConfiguration
VersioningConfiguration	Container for the <i>Status</i> response element.
	Type: Container
	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.

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:xQE0diMbLRepd1
Content-Type: text/plain
```

Sample Response

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

Related Resources

- GET Object
- PUT Object
- DELETE Object

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 <code>LocationConstraint</code> 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 <code>How to Select a Region for Your Buckets</code>.



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-ac1 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 <u>Common Request Headers</u>.

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

Request Elements

Name	Description	Required
CreateBucketConfiguration	Container for bucket configuration settings.	No
	Type: Container	
	Ancestor: None	
LocationConstraint	Specifies the Region where the bucket will be created	No
	Type: Enum	
	Valid Values: EU us-west-1	
	Default: US Standard	
	Ancestor: CreateBucketConfiguration	

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.

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:xQE0diMbLRepd1
```

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQL
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
```

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQL
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:xQE0diMbLRepd1

Sample Response

HTTP/1.1 200 OK

x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQl

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
- DELETE Bucket

PUT Bucket acl

Description

This implementation of the PUT operation uses the *ac1* sub-resource 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-ac1* request header.) To set the ACL of a bucket, you must have WRITE_ACP permission.

For more information about creating a bucket, see

<u>PUT Bucket</u>. For more information about returning the ACL of a bucket, see <u>GET Bucket ACL</u>. For more information about setting the Versioning for a bucket, see <u>PUT Bucket Versioning Status</u>.

Requests

Syntax

```
PUT /?acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
<AccessControlPolicy>
  <0wner>
    <ID>ID</ID>
    <DisplayName>EmailAddress
  </0wner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSc</pre>
        <ID>ID</ID>
        <DisplayName>EmailAddress
      </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.

Request Elements

Name	Description	Required
AccessControlList	Container for Grant, Grantee, and Permission	No
	Type: Container	
	Ancestors: AcessControlPolicy	
AccessControlPolicy	Contains the elements that set the ACL permissions for an object per Grantee.	No
	Type: String	
	Ancestors: None	
DisplayName	Screen name of the bucket owner.	No
	Type: String	
	Ancestors: AcessControlPolicy.Owner	
Grant	Container for the grantee and his or her permissions.	No
	Type: Container	
	Ancestors: AcessControlPolicy.AccessControlList	
Grantee	The subject whose permissions are being set. For more information, see Grantee Values.	No
	Type: String	
	Ancestors: AcessControlPolicy.AccessControlList.Grant	
ID	ID of the bucket owner, or the ID of the grantee.	No
	Type: String	
	Ancestors: AcessControlPolicy.Owner AcessControlPolicy.AccessControlList.Grant	
Owner	Container for the bucket owner's display name and ID.	No
	Type: Container	

	Ancestors: AcessControlPolicy	
Permission	Specifies the permission given to the grantee.	No
	Type: String	
	Valid Values: FULL_CONTROL WRITE WRITE_ACP READ READ_ACP	
	Ancestors: AcessControlPolicy.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:

DisplayName is optional and ignored in the request.

• By E-mail address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchering com/
<EmailAddress>Grantees@email.com/
</Grantee>
```

The grantee is resolved to the *CanonicalUser* and in a response to a GET Object acl request appears as the *CanonicalUser*.

• By URI:

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.

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.

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:xQE0diMbLRepd1
<AccessControlPolicy>
  <0wner>
    <ID>8a6925ce4adf588a4f21c32aa379004fef</ID>
    <DisplayName>BucketOwnersEmail@amazon.com
  </0wner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSc</pre>
        <ID>8a6925ce47f21c32aa379004fef</ID>
        <DisplayName>BucketOwnersEmail@amazon.com</Dis</pre>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample Response

```
HTTP/1.1 200 OK
```

x-amz-id-2: YgIPIfBiKa2bj0KMgUAdQkf3ShJT00pXUueF6QKo x-amz-request-id: 236A8905248E5A01

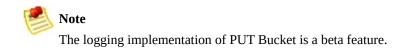
Date: Wed, 01 Mar 2009 12:00:00 GMT

Related Resources

- PUT Bucket
- DELETE Bucket

PUT Bucket logging

Description



This implementation of the PUT operation uses the *logging* sub-resource 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.co"</pre>

For more information about creating a bucket, see

<u>PUT Bucket</u>. For more information about returning the logging status of a bucket, see <u>GET Bucket logging</u>.

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 <u>Common Request Headers</u>.

Request Elements

Name	Description	Required
BucketLoggingStatus	Container for logging status information.	Yes
	Type: Container	
	Children: LoggingEnabled	

	Ancestry: None	
EmailAddress	E-mail address of the person being granted logging permissions.	No
	Type: String	
	Children: None	
	Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant.Grantee	
Grant	Container for the grantee and his/her logging permissions.	No
	Type: Container	
	Children: Grantee, Permission	
	Ancestry: BucketLoggingStatus. LoggingEnabled. TargetGrants	
Grantee	Container for <i>EmailAddress</i> of the person being granted logging permissions. For more information, see <u>Grantee Values</u> .	No
	Type: Container	
	Children: EmailAddress	
	Ancestry: BucketLoggingStatus. LoggingEnabled. TargetGrants. Grant	
LoggingEnabled	Container for logging information. This element is present when you are enabling logging (and not present when you are disabling logging).	No
	Type: Container	
	Children: Grant, TargetBucket, TargetPrefix	
	Ancestry: BucketLoggingStatus	
Permission	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.	No
	Type: String	
	Valid Values: FULL_CONTROL READ WRITE	
	Children: None	
	Ancestry: BucketLoggingStatus.LoggingEnabled.TargetGrants.Grant	
TargetBucket	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.	No
	Type: String	

	Children: None	
	Ancestry: BucketLoggingStatus.LoggingEnabled	
TargetGrants	Container for granting information.	No
	Type: Container	
	Children: Grant, Permission	
	Ancestry: BucketLoggingStatus.LoggingEnabled	
TargetPrefix	This element lets you specify a prefix for the keys that the log files will be stored under.	No
	Type: String	
	Children: None	
	Ancestry: BucketLoggingStatus.LoggingEnabled	

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:

DisplayName is optional and ignored in the request.

• By E-mail address:

```
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchering com/
<EmailAddress>Grantees@email.com/
</Grantee>
```

The grantee is resolved to the *CanonicalUser* and in a response to a GET Object acl request appears as the *CanonicalUser*.

• By URI:

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.

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.

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:xQE0diMbLRepd1
<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.cc</pre>
  <LoggingEnabled>
    <TargetBucket>mybucketlogs</TargetBucket>
    <TargetPrefix>mybucket-access log-/</TargetPrefix>
    <TargetGrants>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XML</pre>
          xsi:type="AmazonCustomerByEmail">
          <EmailAddress>user@company.com</EmailAddress</pre>
        </Grantee>
        <Permission>READ</Permission>
      </Grant>
    </TargetGrants>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQl
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:xQE0diMbLRepd1
<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.co
```

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQl
x-amz-request-id: 236A8905248E5A01
Date: Wed, 01 Mar 2009 12:00:00 GMT
```

Related Resources

- PUT Object
- DELETE Bucket
- PUT Bucket
- GET Bucket logging

PUT Bucket requestPayment

Description

This implementation of the PUT operation uses the requestPayment sub-resource 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

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 <u>Common Request Headers</u>.

Request Elements

Name	Description

Payer	Specifies who pays for the download and request fees.
	Type: Enum
	Valid Values: Requester BucketOwner
	Ancestor: RequestPaymentConfiguration
RequestPaymentConfiguration	Container for Payer.
	Type: Container

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.

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.

Examples

Sample Request

This request creates a Requester Pays bucket named "colorpictures."

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQl
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
- GET Bucket requestPayment

PUT Bucket versioning

Description

This implementation of the PUT operation uses the versioning sub-resource 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

<u>PUT Bucket</u>. For more information about returning the versioning state of a bucket, see <u>GET Bucket Versioning Status</u>.

Requests

Syntax

Note the space between [SerialNumber] and [TokenCode].

Request Parameters

This implementation of the operation does not use request parameters.

Request Headers

Name	Description	Required
x- amz- mfa	The value is the concatenation of the authentication device's serial number, a space, and the value displayed on your authentication device.	Conditional

Type: String

Default: None

Condition: Required to configure the versioning state if versioning is configured with MFA Delete enabled.

Request Elements

Name	Description	Required
Status	Sets the versioning state of the bucket.	No
	Type: Enum	
	Valid Values: Suspended Enabled	
	Ancestor: VersioningConfiguration	
MfaDelete	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.	No
	Type: Enum	
	Valid Values: Disabled Enabled	
	Ancestor: VersioningConfiguration	
	Constraint: Can only be used when you use Status.	
VersioningConfiguration	Container for setting the versioning state.	Yes
	Type: Container	
	Children: Status	
	Ancestor: None	

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.

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.

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:xQE0diMbLRepd1
Content-Type: text/plain
Content-Length: 124
```

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQl
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.

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQl
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:xQE0diMbLRepd1
Content-Type: text/plain
Content-Length: 124
```

Note the space between [SerialNumber] and [TokenCode] and that you must include Status whenever you use MfaDelete.

Sample Response

```
HTTPS/1.1 200 0K
x-amz-id-2: YgIPIfBiKa2bj0KMg95r/0zo3emzU4dzsD4rcKCHQL
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
- PUT Bucket

Operations on Objects

Topics

•

DELETE Object

- GET Object
- GET Object acl
- GET Object torrent
- HEAD Object
- POST Object
- PUT Object
- PUT Object acl
- PUT Object (Copy)

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



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* sub-resource. Using this sub-resource 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 verionId request. Requests that include *x-amz-mfa* must use HTTPS.

For more information about MFA Delete, go to

<u>Using MFA Delete</u>. To see sample requests that use versioning, see <u>Sample Request</u>.

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

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 (true) or was not (false) a delete marker. In a simple DELETE, this header indicates whether (true) or not (false) a delete marker was created.	
	Type: Boolean	
	Valid Values: true false	
	Default: false	
x-amz- version- id	Transfer of the second	
	Type: String	
	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.

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:xQE0diMbLRepd1
Content-Type: text/plain
```

Sample Response

```
HTTP/1.1 204 NoContent
x-amz-id-2: LriYPLdm0dAiIfgSm/F1YsViT1LW94/xUQxMsF7xiE
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=UIORUnfndfiufdisc

Host: bucket.s3.amazonaws.com

Date: Wed, 12 Oct 2009 17:50:00 GMT

Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepd1

Content-Type: text/plain

Content-Length: 0

Sample Response

HTTP/1.1 204 NoContent

x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiE

x-amz-request-id: 0A49CE4060975EAC

x-amz-version-id: UIORUnfndfiufdisojhr398493jfdkjFJjkr

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/xUQxMsF7xiE

x-amz-request-id: 0A49CE4060975EAC

x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+M1

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 HTT Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
x-amz-mfa:[SerialNumber] [AuthenticationCode]
Authorization: AWS 15B4D3461F177624206A:xQE0diMbLRepd1
Content-Type: text/plain
Content-Length: 0
```

Sample Response

```
HTTPS/1.1 204 NoContent
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiE
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
- DELETE Object

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 READ access is granted to the anonymous user, you can return the object without using an authorization header.

If the latest version is a Delete Marker, Amazon S3 behaves as if the object was deleted and Amazon S3 includes in the response: *x-amz-delete-marker: true*.

To distribute large files to many people, you can save bandwidth costs using BitTorrent. For more information, see

Amazon S3 Torrent. For more information about returning the ACL of an object, see GET Object acl.

Versioning

By default, the GET operation returns the latest version of an object. To return a different version, use the *versionId* sub-resource.

For more information about versioning, see <u>PUT Bucket</u> <u>versioning</u>. To see sample requests that use versioning, see <u>Sample Request Getting a Specified Version of an Object</u>.

Requests

Syntax

```
GET / ObjectName 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 can use the following request headers in addition to the request headers common to all operations. For more information, see Common Request Headers.

Name	Description	Required
Range	Downloads the specified range of an object.	No
	Type: String	
	Default: None	
	Constraints: None	
If-Modified-		

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

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 (true) or was not (false) a Delete Marker. If false, this response header does not appear in the response.
marker	Type: Boolean
	Valid Values: true false
	Default: false
x-amz- version-id	Returns the version ID of the retrieved object if it has a unique version ID.
version id	Type: String
	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.

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/cRonhpaE
```

Sample Response

```
HTTP/1.1 200 0K
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9
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 /my0bject?versionId=3/L4kqtJlcpXroDTDmpUMLUo HTTP/Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaE
```

Sample Response to a Versioned Object GET Request

```
HTTP/1.1 200 0K
x-amz-id-2: eftixk72aD6Ap540pIszj7UDNEHGran
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

Connection: close Server: AmazonS3

[434234 bytes of object data]

Related Resources

- <u>GET Service</u>
- GET Object acl

GET Object acl

Description

This implementation of the GET operation uses the *ac1* sub-resource to return the access control list (ACL) of an object. To use this operation, you must have READ_ACP access to the object.

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.

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.

Response Elements

Name	Description
AccessControlList	Container for Grant, Grantee, and Permission
	Type: Container
	Ancestors: AcessControlPolicy
AccessControlPolicy	Contains the elements that set the ACL permissions for an object per Grantee.
	Type: Container
	Ancestors: None
DisplayName	Screen name of the bucket owner
	Type: String
	Ancestors: AcessControlPolicy.Owner
Grant	Container for the grantee and his or her permissions.
	Type: Container
	Ancestors: AcessControlPolicy.AccessControlList
Grantee	The subject whose permissions are being set.
	Type: String
	Ancestors: AcessControlPolicy.AccessControlList.Grant
ID	ID of the bucket owner, or the ID of the grantee
	Type: String

	Ancestors: AcessControlPolicy.Owner or AcessControlPolicy.AccessControlList.Grant
Owner	Container for the bucket owner's display name and ID. Type: Container
	Ancestors: AcessControlPolicy
Permission	Specifies the permission (FULL_CONTROL, WRITE, READ_ACP) given to the grantee.
	Type: String
	Ancestors: AcessControlPolicy.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.

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/cRonhpaE
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 4HL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+M1
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>
  <0wner>
    <ID>8a6925ce4adf588a4532aa379004fef</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </0wner>
  <AccessControlList>
```

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/L4kqtJlcpXroDVBH40Nr8X8cHost: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaE
```

Sample Response Showing the ACL of the Specific Version

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51TnqcoF8eFidJG9Z/2mkiDFu8yU9x-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+M7
```

```
Content-Length: 124
Content-Type: text/plain
Connection: close
Server: AmazonS3
<AccessControlPolicy>
  <0wner>
    <ID>8a6925ce4adf588a421c32aa379004fef</ID>
    <DisplayName>mdtd@amazon.com</DisplayName>
  </0wner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSc</pre>
        <ID>8a6925ce4adf588a4532142d3f74dd8c71fa124b1c
        <DisplayName>mdtd@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Related Resources

- GET Object
- PUT Object
- DELETE 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* sub-resource.

To see sample requests that use Versioning, see <u>Sample</u> Request Getting the ACL of the Specific Version of an <u>Object</u>.

GET Object torrent

Description

This implementation of the GET operation uses the *torrent* sub-resource 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.

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 <u>Common Request Headers</u>.

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.

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.

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/cRonhpaE
```

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.torre
Content-Type: application/x-bittorrent
Content-Length: 537
Server: AmazonS3
<body>
<br/>
<br
```

Related Resources

• GET Object

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. If READ access is granted to the anonymous user, you can request the object's metadata without an authorization header.

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.

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.

Name	Description	Required
Range	Downloads the specified range of an object.	No
	Type: String	
	Default: None	
	Constraints: None	

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

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.

Name	Description
x-amz- meta-*	If you supplied user metadata when you PUT the object, that metadata is returned in one or more response headers prefixed with x-amz-meta- and with the suffix name that you provided on storage, for example, family, i.e., x-amz-meta-family. Amazon S3 returns this metadata verbatim; Amazon S3 does not interpreted it. Type: String
x-amz- missing- meta	This is set to the number of metadata entries not returned in x-amz-meta headers. This can happen if you create metadata using an API like SOAP that supports more flexible metadata than the REST API. For example, using SOAP, you can create metadata whose values are not legal HTTP headers. Type: String
x-amz- version- id	This is set to the version ID of the object returned. 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.

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: 3HL4kqtJlcpXroDTDmjVBH4ONrjfkd
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=3HL4kgCxf3vjVBH40Nrjfkd F
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

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 <code>versionId</code> sub-resource. For more information, see <code>Versions</code> in the <code>Amazon Simple Storage Service Developer Guide</code>.

To see sample requests that use versioning, see <u>Sample</u> Request Getting Metadata From a Specified Version of an <u>Object</u>.

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. 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 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 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.

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.

Requests

Syntax

```
POST /ObjectName?acl HTTP/1.1
Host: <a href="mailto:destinationBucket">destinationBucket</a>.s3.amazonaws.com
User-Agent: browser_data
Accept: file_types
Accept-Language: Regions
Accept-Encoding: <a href="mailto:encoding">encoding</a>
Accept-Charset: character set
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156
Content-Length: length
--9431149156168
Content-Disposition: form-data; name="key"
filepath
--9431149156168
Content-Disposition: form-data; name="acl"
ac1
--9431149156168
Content-Disposition: form-data; name="success action r
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=
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.

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.

Name	Description	Required
AWSAccessKeyId	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.	Conditional
	Type: String	
	Default: None	
	Constraints: Required if a policy document is included with the request.	
acl	Specifies an Amazon S3 access control list. If an invalid access control list is specified, an error is generated. For more information on ACLs, go to <u>Amazon S3 ACLs</u> in the <u>Amazon Simple Storage Service Developer Guide</u> .	No
	Type: String	
	Default: private	
	Valid Values: private public-read public-read-write authenticated-read bucket-owner-read bucket-owner-full-control	
Cache-Control, Content- Type, Content-	REST-specific headers.	No
Disposition, Content- Encoding	For more information, see <u>PUT Object</u> .	
	Type: String	
	Default: None	
expires	Number of milliseconds before expiration	No
	Type: Int	
	Default: None	
file	File or text content.	Yes

	The file or text content must be the last field in the form.	
	You cannot upload more than one file at a time.	
	Type: File or text content	
	Default: None	
key	The name of the uploaded key.	Yes
	To use the filename provided by the user, use the \${filename} variable. For example, if the user Betty uploads the file the file lolcatz.jpg and you specify /user/betty/\${filename}, the file will be stored as /user/betty/lolcatz.jpg.	
	For more information, go to <u>Using Keys</u> in the <u>Amazon Simple</u> <u>Storage Service Developer Guide</u> .	
	Type: String	
	Default: None	
policy	Security Policy describing what is permitted in the request. Requests without a security policy are considered anonymous and only work on publicly writable buckets.	No
	Type: String	
	Default: None	
success_action_redirect, redirect	The URL to which the client is redirected upon successful upload.	No
	If success_action_redirect is not specified, Amazon S3 returns the empty document type specified in the success_action_status field.	
	If Amazon S3 cannot interpret the URL, it acts as if the field is not present.	
	If the upload fails, Amazon S3 displays an error and does not redirect the user to a URL.	
	Type: String	
	Default: None	
	Note	
	The redirect field name is	
	deprecated and support for the redirect field name will be removed in the future.	

success_action_status	The status code returned to the client upon successful upload if success_action_redirect is not specified.	No
	Accepts the values 200, 201, or 204 (default).	
	If the value is set to 200 or 204, Amazon S3 returns an empty document with a 200 or 204 status code.	
	If the value is set to 201, Amazon S3 returns an XML document with a 201 status code.	
	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.	
	Type: String	
	Default: None	
	Note	
	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	
	success_action_status to 201.	
x-amz-meta-*	Field names prefixed with <i>x-amz-meta-</i> contain user-specified	No
	metadata.	
	Amazon S3 does not validate or use this data.	
	For more information, see <u>PUT Object</u> .	
	Type: String	
	Default: None	
x-amz-security-token	Amazon DevPay security token.	No
	Each request that uses Amazon DevPay requires two x-amz-security-token form fields: one for the product token and one for the user token.	
	For more information, go to <u>Using DevPay</u> .	

Type: String	
Default: None	

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.

Name	Description
success_action_redirect, redirect	The URL to which the client is redirected on successful upload.
	Type: String
	Ancestor: PostResponse
x-amz-version-id	Version of the object.
	Type: String

Response Elements

Name	Description
Bucket	Name of the bucket the object was stored in.
	Type: String
	Ancestor: PostResponse
ETag	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.
	Type: String
	Ancestor: PostResponse
Key	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.
	Type: String
	Ancestor: PostResponse
Location	URI of the object.
	Type: String
	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.

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:xQE0diMbLRepd1
Content-Type: text/plain
Expect: the 100-continue HTTP status code
```

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 0K
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiE
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/xUQxMsF7xiE
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fjj3HHNVfdsQUIFDNsDate: Wed, 01 Mar 2009 12:00:00 GMT
ETag: "828ef3fdfa96f00ad9f27c383fc9ac7f"
Content-Length: 0
Connection: close
Server: AmazonS3
```

Related Resources

- PUT Object (Copy)
- POST Object
- GET Object

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 <u>GET Bucket (Versioning Status)</u>.

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 <u>Sample</u> <u>Request</u>.

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 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.

Requests

Syntax

PUT / ObjectName? 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 can use the following request headers in addition to the request headers common to all operations. For more information, see Common Request Headers.

Name	Description	Required
Cache- Control	Can be used to specify caching behavior along the request/reply chain. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9 . Type: String Default: None Constraints: None	No

Content- Disposition	Specifies presentational information for the object. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1 . Type: String Default: None Constraints: None	No
Content- Encoding	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 <code>Content-Type</code> header field. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11 . Type: String Default: None Constraints: None	No
Content- Length	The size of the object, in bytes. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13 . Type: String Default: None Constraints: None	Yes
Content-MD5	The base64 encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, go to REST Authentication in the Amazon Simple Storage Service Developer Guide Type: String Default: None Constraints: None	No
Content- Type	A standard MIME type describing the format of the contents. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17 . Type: String Default: binary/octet-stream Valid Values: MIME types Constraints: None	No
Expect	When your application uses 100-continue, it does not send the request body until it receives an acknowledgement. If the message is rejected based on the headers, the body of the message is not sent.	No

	Type: String	
	Default: None	
	Valid Values: 100-continue	
	Constraints: None	
Expires	Number of milliseconds before expiration	No
	Type: Int	
	Default: None	
	Constraints: None	
x-amz-acl	The canned ACL to apply to the object. For more information, go to <u>REST</u> <u>Access Policy</u> in the <u>Amazon Simple Storage Service Developer Guide</u>	No
	Type: String	
	Default: private	
	Valid Values: private public-read public-read-write authenticated-read bucket-owner-read bucket-owner-full-control	
	Constraints: None	
x-amz-meta-	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 total size of the HTTP request, not including the body, must be less than 8 KB.	No
	Type: String	
	Default: None	
	Constraints: None	

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.

Name	Description
x-amz-version-id	Version of the object.
	Type: String

Response Elements

This implementation of the operation does not return response elements.

Special Errors

This implementation of the operation does not return special errors. For general information about Amazon S3 errors and a list of error codes, see Error Responses.

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:xQE0diMbLRepd1
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 0K
x-amz-id-2: LriYPLdmOdAiIfgSm/F1YsViT1LW94/xUQxMsF7xiE
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 0K
x-amz-id-2: LriYPLdm0dAiIfgSm/F1YsViT1LW94/xUQxMsF7xiE
x-amz-request-id: 0A49CE4060975EAC
x-amz-version-id: 43jfkodU8493jnFJD9fjj3HHNVfdsQUIFDNsDate: 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.

Related Resources

- PUT Object (Copy)
- POST Object
- GET Object

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 Status).

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 <u>Sample</u> <u>Request</u>.

PUT Object acl

Description

This implementation of the PUT operation uses the *ac1* sub-resource 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-ac1* 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* sub-resource.

To see sample requests that use versioning, see

Sample Request Setting the ACL of a Specified Object Version.

Requests

Syntax

```
PUT / ObjectName? acl HTTP/1.1
Host: BucketName.s3.amazonaws.com
Date: date
Authorization: signatureValue
<AccessControlPolicy>
  <0wner>
    <ID>ID</ID>
    <DisplayName>EmailAddress
  </0wner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSc</pre>
        <ID>ID</ID>
        <DisplayName>EmailAddress
      </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 can use the following request headers in addition to the request headers common to all operations. For more information, see Common Request Headers.

Name	Description	Required
Cache- Control	Can be used to specify caching behavior along the request/reply chain. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9 .	No
	Type: String	
	Default: None	
	Constraints: None	
Content- Disposition	Specifies presentational information for the object. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1 .	No
	Type: String	
	Default: None	
	Constraints: None	
Content- Encoding	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 <code>Content-Type</code> header field. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11 .	No
	Type: String	
	Default: None	
	Constraints: None	
Content- Length	The size of the object, in bytes. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13 .	Yes
	Type: String	
	Default: None	
	Constraints: None	
Content-MD5	The base64 encoded 128-bit MD5 digest of the message (without the headers) according to RFC 1864. This header can be used as a message integrity check to verify that the data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information about REST request authentication, go to	No

	REST Authentication in the Amazon Simple Storage Service Developer Guide .	
	Type: String	
	Default: None	
	Constraints: None	
Content- Type	A standard MIME type describing the format of the contents. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.17 .	No
	Type: String	
	Default: binary/octet-stream	
	Valid Values: MIME types	
	Constraints: None	
Expect	When your application uses 100-continue, it does not send the request body until it receives an acknowledgement. If the message is rejected based on the headers, the body of the message is not sent.	No
	Type: String	
	Default: None	
	Valid Values: 100-continue	
	Constraints: None	
Expires	Number of milliseconds before expiration	No
	Type: Int	
	Default: None	
	Constraints: None	
x-amz-acl	The canned ACL to apply to the object. For more information, go to <u>REST Access Policy</u> in the <u>Amazon Simple Storage Service Developer Guide</u> .	No
	Type: String	
	Default: private	
	Valid Values: private public-read public-read-write authenticated-read bucket-owner-read bucket-owner-full-control	
	Constraints: None	
x-amz-meta-	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 total size of the HTTP request, not including the body, must be less than 8 KB.	No

Default: None	
Constraints: None	

Request Elements

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

Name	Description
AccessControlList	Container for ACL information
	Type: Container
	Ancestors: AcessControlPolicy
AccessControlPolicy	Contains the elements that set the ACL permissions for an object per Grantee
	Type: Container
	Ancestors: None
DisplayName	Screen name of the bucket owner
	Type: String
	Ancestors: AcessControlPolicy.Owner
Grant	Container for the grantee and his or her permissions
	Type: Container
	Ancestors: AcessControlPolicy.AccessControlList
Grantee	The subject whose permissions are being set.
	Type: String
	Valid Values: DisplayName EmailAddress AuthenticatedUser. For more information, see <u>Grantee Values</u> .
	Ancestors: AcessControlPolicy.AccessControlList.Grant
ID	ID of the bucket owner, or the ID of the grantee
	Type: String
	Ancestors: AcessControlPolicy.Owner or AcessControlPolicy.AccessControlList.Grant

Owner	Container for the bucket owner's display name and ID
	Type: Container
	Ancestors: AcessControlPolicy
Permission	Specifies the permission given to the grantee
	Type: String
	Valid Values: FULL_CONTROL WRITE WRITE_ACP READ READ_ACP
	Ancestors: AcessControlPolicy.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/XMLSchering">
    <ID>ID</ID>
    <DisplayName>GranteesEmail</DisplayName>
    </Grantee>
```

DisplayName is optional and ignored in the request.

• By E-mail address:

The grantee is resolved to the *CanonicalUser* and in a response to a GET Object acl request appears as the *CanonicalUser*.

• By URI:

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.

Name	Description
x-amz-version-id	Version of the object whose ACL is being set.
	Type: String
	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.

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/cRonhpaE
Content-Length: 124
<AccessControlPolicy>
  <0wner>
    <ID>8a6925ce4adf588e97f21c32aa379004fef</ID>
    <DisplayName>CustomersName@amazon.com</DisplayName</pre>
  </0wner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSc</pre>
        <ID>8a6925ce4adf588a45379004fef</ID>
        <DisplayName>CustomerName@amazon.com
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample Response

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

```
HTTP/1.1 200 OK
x-amz-id-2: eftixk72aD6Ap51T9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 3/L4kqtJlcpXrof3vjVBH40Nr8X8gdRQBpUN
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.

Sample Response

```
HTTP/1.1 200 OK x-amz-id-2: eftixk72aD6Ap51u8yU9AS1ed4OpIszj7UDNEHGrar x-amz-request-id: 318BC8BC148832E5 x-amz-version-id: 3/L4kqtJlcpXro3vjVBH40Nr8X8gdRQBpUML 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)
- POST Object
- GET Object

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.

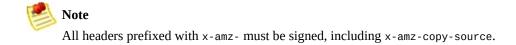
When copying an object, you can preserve most of the metadata (default) or specify new metadata. However, the ACL is not preserved and is set to private for the user making the request. To override the default ACL setting, use the *x-amz-ac1* header to specify a new ACL when generating a copy request. 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.



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.



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.

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 <u>Common Request Headers</u>.

Name	Description	Required
x-amz-acl	The canned ACL to apply to the object. For more information, go to <u>REST Access Policy</u> in the <u>Amazon Simple Storage Service Developer Guide</u> .	No
	Type: String	
	Default: private	
	Valid Values: private public-read public-read-write authenticated-read bucket-owner-read bucket-owner-full-control	
	Constraints: None	
x-amz-copy- source	The name of the source bucket and key name of the source object, separated by a slash (/).	Yes
	Type: String	
	Default: None	
	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.	
x-amz- metadata- directive	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.	No
	Type: String	
	Default: COPY	
	Valid values: COPY REPLACE	
	Constraints: Values other than COPY or REPLACE result in an immediate 400-based error response. You cannot copy an object to itself unless the MetadataDirective header is specified and its value set to REPLACE.	
	For information on supported metadata, see <u>Common Request Headers</u>	
x-amz-copy- source-if- match	Copies the object if its entity tag (ETag) matches the specified tag; otherwise, the request returns a 412 HTTP status code error (precondition failed).	No
	Type: String	
	Default: None	
	Constraints: This header can be used with x-amz-copy-source-if-unmodified-since, but cannot be used with other conditional copy headers.	
x-amz-copy- source-if- none-match	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).	No
	Type: String	
	Default: None	

	Constraints: This header can be used with x-amz-copy-source-if-modified-since, but cannot be used with other conditional copy headers.	
x-amz-copy- source-if- unmodified- since	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). Type: String Default: None Constraints: This must be a valid HTTP date. For more information, go to http://www.ietf.org/rfc/rfc2616.txt . This header can be used with x-amz-copy-	No
x-amz-copy- source-if-	source-if-match, but cannot be used with other conditional copy headers. Copies the object if it has been modified since the specified time; otherwise, the request returns a 412 HTTP status code error (failed condition).	No
modified- since	Type: String Default: None	
	Constraints: This must be a valid HTTP date. This header can be used with x-amz-copy-source-if-none-match, but cannot be used with other conditional copy headers.	

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.

Name	Description
x-amz-copy-source-version-id	Version of the source object that was copied.
	Type: String
x-amz-version-id	Version of the copied object in the destination bucket.
	Type: String

Response Elements

Description
Container for all response elements.
Type: Container
Ancestor: None
Returns the ETag of the new object. The ETag only reflects changes to the contents of an object, not its metadata.
Type: String
Ancestor: CopyObjectResult
Returns the date the object was last modified.
Type: String

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.

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/cRonhpaE
```

Sample Response

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/L4
Authorization: AWS 02236Q3V0WHVSRW0EXG2:0RQf4/cRonhpaE
```

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/2mkiDFu8yU9
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: QUpfdndhfd8438MNFDN93jdnJFkdmqnh893
x-amz-copy-source-version-id: 09df8234529fjs0dfi0w529
Date: Wed, 28 Oct 2009 22:32:00 GMT
Connection: close
```

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.

Related Resources

- Copying Objects
- PUT Object
- GET Object

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* sub-resource.

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 <u>Sample</u> Request Copying a Specified Version of an Object.

SOAP API

Topics

- Operations on the Service
- Operations on Buckets
- Operations on Objects

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

Operations on the Service

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/2
    <AWSAccessKeyId>1D9FVRAYCP1VJEXAMPLE=</AWSAccessKeyI
    <Timestamp>2006-03-01T12:00:00.183Z</Timestamp>
     <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</ListAllMyBuckets>
```

Sample Response

</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.

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

- DeleteBucket
- ListBucket
- GetBucketAccessControlPolicy
- <u>SetBucketAccessControlPolicy</u>
- GetBucketLoggingStatus
- <u>SetBucketLoggingStatus</u>

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

Sample Response

<CreateBucketResponse xmlns="http://s3.amazonaws.com/c</pre>

```
<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.

You must authenticate with a valid AWS Access Key ID. Anonymous requests are never allowed to create buckets.

Related Resources

• <u>ListBucket</u>

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

Sample Response

Elements

• *Bucket:* The name of the bucket you want to delete.

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 <u>Listing Keys</u>.

Requests

This example lists up to 1000 keys in the "quotes" bucket that have the prefix "notes."

Syntax

Parameters

Name	Description	Required
prefix	Limits the response to keys which begin with the indicated prefix. You can use prefixes to separate a bucket into different sets of keys in a way similar to how a file system uses folders. Type: String	No
	Default: None	
marker	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.	No
	Type: String	
	Default: None	

max-keys	The maximum number of keys you'd like to see in the response body. The server might return fewer than this many keys, but will not return more. Type: String Default: None	No
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 CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response. Type: String Default: None	No

Success Response

This response assumes the bucket contains the following keys:

```
notes/todos.txt
notes/2005-05-23/customer_mtg_notes.txt
notes/2005-05-23/phone_notes.txt
notes/2005-05-28/sales_notes.txt
```

Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2</pre>
  <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/LastModifi
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;
    <Size>5126</Size>
    <StorageClass>STANDARD</StorageClass>
    <0wner>
      <ID>bcaf1ffd86f41ce161ca5fb16fd081034f</ID>
      <DisplayName>webfile/DisplayName>
    </0wner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
```

```
<Prefix>notes/2005-05-23/</prefix>
</CommonPrefixes>
<CommonPrefixes>
  <Prefix>notes/2005-05-28/</prefix>
</CommonPrefixes>
</ListBucketResult>
```

As you can see, many of the fields in the response echo the request parameters. *IsTruncated*, *Contents*, and *CommonPrefixes* are the only response elements that can contain new information.

Response Elements

Name	Description
Contents	Metadata about each object returned.
	Type: XML metadata
	Ancestor: ListBucketResult
CommonPrefixes	A response can contain <code>commonPrefixes</code> only if you specify a <code>delimiter</code> . When you do, <code>commonPrefixes</code> contains all (if there are any) keys between <code>Prefix</code> and the next occurrence of the string specified by <code>delimiter</code> . In effect, <code>commonPrefixes</code> lists keys that act like subdirectories in the directory specified by <code>Prefix</code> . For example, if <code>prefix</code> is notes/ and <code>delimiter</code> is a slash (/), in notes/summer/july, the common prefix is <code>notes/summer/</code> .
	Type: String
	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 CommonPrefixes collection. These rolled-up keys are not returned elsewhere in the response.
	Type: String
	Ancestor: ListBucketResult
IsTruncated	Specifies whether (true) or not (false) all of the results were returned. All of the results may not be returned if the number of results exceeds that specified by <code>MaxKeys</code> .

	Type: String
	Ancestor: boolean
Marker	Indicates where in the bucket to begin listing.
	Type: String
	Ancestor: ListBucketResult
MaxKeys	The maximum number of keys returned in the response body.
	Type: String
	Ancestor: ListBucketResult
Name	Name of the bucket.
	Type: String
	Ancestor: ListBucketResult
Prefix	Keys that begin with the indicated prefix.
	Type: String
	Ancestor: ListBucketResult

Response Body

For information about the list response, see <u>Listing Keys</u> <u>Response</u>.

To list the keys of a bucket you need to have been granted READ access on the bucket.

GetBucketAccessControlP

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

Sample Response

Response Body

The response contains the access control policy for the bucket. For an explanation of this response, see SOAP Access Policy .

You must have READ_ACP rights to the bucket in order to retrieve the access control policy for a bucket.

SetBucketAccessControlPc

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

Sample Response

You must have WRITE_ACP rights to the bucket in order to set the access control policy for a bucket.

GetBucketLoggingStatus



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 <u>Server Logs</u>. For information about the response document, see <u>Logging API</u>.

Example

Sample Request

Sample Response

Only the owner of a bucket is permitted to invoke this operation.

SetBucketLoggingStatus



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 <u>Server Logs</u>. For information about the response document, see <u>Logging API</u>.

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

Sample Response

Access Control

Only the owner of a bucket is permitted to invoke this operation.

Operations on Objects

Topics

•

PutObjectInline

- PutObject
- CopyObject
- GetObject
- GetObjectExtended
- <u>DeleteObject</u>
- <u>GetObjectAccessControlPolicy</u>
- SetObjectAccessControlPolicy

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 <code>InlineDataTooLargeError</code> 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

```
<Put0bjectInline xmlns="http://doc.s3.amazonaws.com/26</pre>
 <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=
 <ContentLength>5</ContentLength>
 <AccessControlList>
   <Grant>
      <Grantee xsi:type="CanonicalUser">
       <ID>a9a7b886d6fde241bf9b1c61be666e9</ID>
       <DisplayName>chriscustomer
      </Grantee>
      <Permission>FULL CONTROL</Permission>
   </Grant>
   <Grant>
```

Sample Response

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 pre-existing 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

- <u>PutObject</u>
- CopyObject

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

```
<Put0bject xmlns="http://doc.s3.amazonaws.com/2006-03-</pre>
  <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
      </Grantee>
      <Permission>FULL CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/Al
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>1D9FVRAYCP1VJEXAMPLE=</AWSAccessKey]
  <Timestamp>2007-05-11T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=
</Put0bject>
```

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 pre-existing 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

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 <u>Using Auth Access</u>.

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.



You might need to configure the SOAP stack socket timeout for copying large objects.

Request Syntax

```
<CopyObject xmlns="http://bucket name.s3.amazonaws.com")</pre>
  <SourceBucket>source bucket/SourceBucket>
  <SourceObject>source_object/SourceObject>
  <DestinationBucket>destination bucket/DestinationBucker
  <DestinationObject>destination object/DestinationObject
  <MetadataDirective>{REPLACE | COPY}</MetadataDirecti</pre>
  <Metadata>
    <Name>metadata name</Name>
    <Value>metadata value</Value>
  </Metadata>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="user_type">
        <ID>user id</ID>
        <DisplayName>display_name
      </Grantee>
      <Permission>permission</Permission>
    </Grant>
  </AccessControlList>
  <CopySourceIfMatch>etag</CopySourceIfMatch>
  <CopySourceIfNoneMatch>etag</CopySourceIfNoneMatch>
  <CopySourceIfModifiedSince>date_time/CopySourceIfMod:
  <CopySourceIfUnmodifiedSince>date_time</CopySourceIfUr
  <AWSAccessKeyId>AWSAccessKeyId</AWSAccessKeyId>
  <Timestamp>TimeStamp</Timestamp>
  <Signature>Signature</Signature>
</CopyObject>
```

Request Parameters

Name	Description	Required
SourceBucket	The name of the source bucket.	Yes
	Type: String	
	Default: None	
	Constraints: A valid source bucket.	
SourceKey	The key name of the source object.	Yes
	Type: String	
	Default: None	
	Constraints: The key for a valid source object to which you have READ access.	
DestinationBucket	The name of the destination bucket.	Yes
	Type: String	
	Default: None	
	Constraints: You must have WRITE access to the destination bucket.	
DestinationKey	The key of the destination object.	Yes
	Type: String	
	Default: None	
	Constraints: You must have WRITE access to the destination bucket.	
MetadataDirective	Specifies whether the metadata is copied from the source object or replaced with metadata provided in the request.	No
	Type: String	
	Default: COPY	
	Valid values: COPY REPLACE	
	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.	

Metadata	Specifies metadata name-value pairs to set for the object.If MetadataDirective is set to COPY, all metadata is ignored. Type: String Default: None Constraints: None.	No
AccessControlList	Grants access to users by e-mail addresses or canonical user ID. Type: String Default: None Constraints: None	No
CopySourceIfMatch	Copies the object if its entity tag (ETag) matches the specified tag; otherwise return a PreconditionFailed. Type: String Default: None Constraints: None. If the Etag does not match, the object is not copied.	No
CopySourceIfNoneMatch	Copies the object if its entity tag (ETag) is different than the specified Etag; otherwise returns an error. Type: String Default: None Constraints: None.	No
CopySourceIfUnmodifiedSince	Copies the object if it hasn't been modified since the specified time; otherwise returns a PreconditionFailed. Type: dateTime Default: None	No
CopySourceIfModifiedSince	Copies the object if it has been modified since the specified time; otherwise returns an error. Type: dateTime Default: None	No

Response Syntax

Response Elements

Name	Description
Etag	Returns the etag of the new object. The ETag only reflects changes to the contents of an object, not its metadata.
	Type: String
	Ancestor: CopyObjectResult
LastModified	Returns the date the object was last modified.
	Type: String
	Ancestor: CopyObjectResult

For information about general response elements, see Using REST Error Response Headers.

Special Errors

There are no special errors for this operation. For information about general Amazon S3 errors, see <u>List of Error Codes</u>.

Examples

This example copies the flotsam object from the pacific bucket to the jetsam object of the atlantic bucket, preserving its metadata.

Sample Request

Sample Response

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</pre>
  <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=</AWSAccessKey]</pre>
  <Timestamp>2008-02-18T13:54:10.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbq7RrtSFmw=</Signature>
</CopyObject>
```

Sample Response

Related Resources

- Copying Objects
- <u>PutObject</u>
- <u>PutObjectInline</u>

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 operation.

Example

This example gets the "Nelson" object from the "quotes" bucket.

Sample Request

Sample Response

```
<GetObjectResponse xmlns="http://s3.amazonaws.com/doc/</pre>
  <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/LastModifi
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;
  </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
- For GET operations that fail, you can design your application to download the remainder instead of the entire file. For more information, see <u>REST GET</u> <u>Error Recovery</u>

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.

```
<Bucket>bigbucket
<Prefix>bigfile</prefix>
  <MaxKeys>1</MaxKeys>
  <AWSAccessKeyId>1D9FVRAYCP1VJEXAMPLE=</AWSAccessKey]
  <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/2</pre>
  <Name>quotes</Name>
  <Prefix>N</Prefix>
  <MaxKeys>1</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>bigfile</Key>
    <LastModified>2006-01-01T12:00:00.000Z/LastModifi
    <ETag>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;
    <Size>2023276</Size>
    <StorageClass>STANDARD</StorageClass>
    <0wner>
      <ID>bcaf1ffd86f41161ca5fb16fd081034f</ID>
      <DisplayName>bigfile</DisplayName>
     </0wner>
  </Contents>
</ListBucketResult>
```

Following is a request that downloads the first megabyte from the bigfile object.

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

```
<Bucket>bigbucket</Bucket>
  <Key>bigfile</Key>
  <GetMetadata>true</GetMetadata>
  <GetData>true</GetData>
  <InlineData>true</InlineData>
  <ByteRangeStart>0</ByteRangeStart>
  <ByteRangeEnd>1048576</ByteRangeEnd>
  <AWSAccessKeyId>1D9FVRAYCP1VJEXAMPLE=</AWSAccessKeyI
  <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/</pre>
  <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/LastModifi
    <ETag>"828ef3fdfa96f00ad9f27c383fc9ac7f"</ETag>
  </GetObjectResponse>
</GetObjectResponse>
```

To ensure the file did not change since the previous portion was downloaded, specify the IfMatch element. Although the IfMatch element is not required, it is recommended for content that is likely to change.

The following is a request that gets the remainder of the file, using the IfMatch request header.

Amazon S3 returns the following response and the remainder of the file.

Versioned GetObject

The following request returns the specified version of the object in the bucket.

```
<GetObject xmlns="http://doc.s3.amazonaws.com/2006-03-
<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/</pre>
<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>&quot;828ef3fdfa96f00ad9f27c383fc9ac7f&quot;</El
</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.

Related Resources

Operations on Objects

GetObjectExtended

GetObjectExtended is exactly like GetObject, 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: Return 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.



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

Sample Response

<DeleteObjectResponse xmlns="http://s3.amazonaws.com/c</pre>

```
<DeleteObjectResponse>
     <Code>200</Code>
     <Description>0K</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.

GetObjectAccessControlPo

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

Sample Response

```
<AccessControlPolicy>
    <0wner>
        <ID>a9a7b886d6fd24a541bf9b1c61be666e9</ID>
        <DisplayName>chriscustomer</DisplayName>
        </0wner>
        <AccessControlList>
        <Grant>
                <Grantee xsi:type="CanonicalUser">
```

Response Body

The response contains the access control policy for the bucket. For an explanation of this response, <u>SOAP</u>
<u>Access Policy</u>.

Access Control

You must have READ_ACP rights to the object in order to retrieve the access control policy for an object.

SetObjectAccessControlPo

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>

Sample Response

Access Control

You must have WRITE_ACP rights to the object in order to set the access control policy for a bucket.

Glossary

100-continue

A method that enables a client to see if a server can accept a request before actually sending it. For large PUTs, this can save both time and bandwidth charges.

account

AWS account associated with a particular developer.

authentication

The process of proving your identity to the system.

bucket

A container for objects stored in Amazon S3. Every object is contained within a bucket. For example, if the object named photos/puppy.jpg is stored in the johnsmith bucket, then it is addressable using the URL

http://johnsmith.s3.amazonaws.com/photos/pup

canned access policy

A standard access control policy that you can apply to a bucket or object. Valid Values: private | public-read | public-read-write | authenticated-read | bucket-owner-read | bucket-owner-full-control

canonicalization

The process of converting data into a standard format that will be recognized by a service such as Amazon S3.

consistency model

The method through which Amazon S3 achieves high availability, which involves replicating data across multiple servers within Amazon's data centers. After a "success" is returned, your data is safely stored. However, information about the changes might not immediately replicate across Amazon S3.

key

The unique identifier for an object within a bucket. Every object in a bucket has exactly one key. Since a bucket and key together uniquely identify each object, Amazon S3 can be thought of as a basic data map between "bucket + key" and the object itself. Every object in Amazon S3 can be uniquely addressed through the combination of the web service endpoint, bucket name, and key, as in

http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl, where "doc" is the name of the bucket, and "2006-03-01/AmazonS3.wsdl" is the key.

metadata

The metadata is a set of name-value pairs that describe the object. These include default metadata such as the date last modified and standard HTTP metadata such as Content-Type. The developer can also specify custom metadata at the time the Object is stored.

object

The fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3.

service endpoint

The host and port with which you are trying to communicate within the destination URL. For virtual hosted-style requests, this is mybucket.s3.amazonaws.com. For path-style requests, this is s3.amazonaws.com

Document Conventions

This section lists the common typographical and symbol use conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical use conventions.

Convention	Description/Example		
Call-outs	A call-out is a number in the body text to give you a visual reference. The reference point is for further discussion elsewhere.		
	You can use this resource regularly.		
Code in text	Inline code samples (including XML) and commands are identified with a special font.		
	You can use the command java -version.		
Code blocks	Blocks of sample code are set apart from the body and marked accordingly.		
	<pre># ls -l /var/www/html/index.html -rw-rw-r 1 root root 1872 Jun 21 09:33 /var/www/html/index.htm # date Wed Jun 21 09:33:42 EDT 2006</pre>		
Emphasis	Unusual or important words and phrases are marked with a special font.		
	You <i>must</i> sign up for an account before you can use the service.		
Internal cross	References to a section in the same document are marked.		
references	For more information, see <u>Document Conventions</u> .		
Logical	A special font is used for expressions that are important to identify, but are not code.		
values, constants, and regular expressions, abstracta	If the value is null, the returned response will be false.		
Product and	Named AWS products and features are identified on first use.		
feature names	Create an Amazon Machine Image (AMI).		
Operations	In-text references to operations.		
	Use the GetHITResponse operation.		
Parameters	In-text references to parameters.		

	The operation accepts the parameter <i>AccountID</i> .	
Response elements	In-text references to responses.	
	A container for one CollectionParent and one or more CollectionItems.	
Technical publication references	References to other AWS publications. If the reference is hyperlinked, it is also underscored.	
	For detailed conceptual information, refer to the <i>Amazon Mechanical Turk Developer Guide</i> .	
User entered values	A special font marks text that the user types.	
	At the password prompt, type MyPassword.	
User interface controls and labels	Denotes named items on the UI for easy identification.	
	On the File menu, click Properties .	
Variables	When you see this style, you must change the value of the content when you copy the text of a sample to a command line.	
	% ec2-register < <i>your-s3-bucket</i> >/image.manifest	
	See also the following symbol convention.	

Symbol Conventions

This section describes the common use of symbols.

Symbol	Description/Example
(Parentheses and vertical bars)	Within a code description, bar separators denote options from which one must be chosen.
	% data = hdfread (start stride edge)
[square brackets]	Within a code description, square brackets denote completely optional commands or parameters.
	% sed [-n, -quiet]
	Use square brackets in XML examples to differentiate them from tags.
	<customerid>[ID]</customerid>
<arrow brackets=""></arrow>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value.
	% ec2-register <your-s3-bucket>/image.manifest</your-s3-bucket>
	(Parentheses and vertical bars) [square brackets]

Index

```
Α
API
   REST, REST API
audience, Who Should Read this Guide
B
buckets
   creating, CreateBucket
   putting, PUT Bucket, PUT Bucket acl, PUT Bucket
   logging, PUT Bucket versioning
   REST operations, Operations on Buckets
   SOAP operations, Operations on Buckets
\mathbf{C}
changes, What's New
common response headers, Common Response Headers
content-MD5, POST Object, PUT Object, PUT Object
acl, PutObject
Content-MD5, PutObjectInline
COPY, REST, PUT Object (Copy)
copying objects
```

REST, PUT Object (Copy)

```
SOAP, CopyObject
CopyObject
   errors, Special Errors
CopyObject, SOAP, CopyObject
CreateBucket
   SOAP, CreateBucket
creating buckets
   SOAP, CreateBucket
D
data corruption
   preventing, POST Object, PUT Object, PUT Object
   acl, PutObjectInline, PutObject
DELETE
   REST, DELETE Bucket, DELETE Object
DeleteBucket, DeleteBucket
deleting objects
   SOAP, DeleteObject
duplicating objects, CopyObject
E
errors
   CopyObject, Special Errors
   list, List of Error Codes
```

```
GET
   policy, REST, GET Service
   REST, GET Bucket (List Objects), GET Bucket
   Object versions, GET Bucket requestPayment, GET
   Bucket versioning, GET Object acl
GET, REST, GET Bucket acl, GET Bucket location, GET
Bucket logging, GET Object, GET Object torrent
GetBucketAccessControlPolicy
   SOAP, GetBucketAccessControlPolicy
GetBucketLoggingStatus
   SOAP, GetBucketLoggingStatus
GetObject
   SOAP, GetObject
GetObjectAccessControlPolicy
   SOAP, GetObjectAccessControlPolicy
GetObjectExtended
   SOAP, GetObjectExtended
glossary, Glossary
guide organization, How This Guide Is Organized
H
HEAD object
```

REST, HEAD Object

headers, request

REST, Common Request Headers

```
T,
ListAllMyBuckets
   SOAP, ListAllMyBuckets
ListBucket
   SOAP, ListBucket
\mathbf{O}
objects
   copying, PUT Object (Copy), CopyObject
   REST operations, Operations on Objects
   SOAP operations, Operations on Objects
organization of guide, How This Guide Is Organized
P
POST, REST, POST Object
PUT
   REST, PUT Bucket, PUT Bucket acl, PUT Bucket
   logging, PUT Bucket requestPayment, PUT Bucket
   versioning
PUT, REST, PUT Object, PUT Object acl
PutObject
   SOAP, PutObject
```

PutObjectInline

SOAP, <u>PutObjectInline</u>
putting buckets

REST, <u>PUT Bucket</u>, <u>PUT Bucket acl</u>, <u>PUT Bucket logging</u>, <u>PUT Bucket versioning</u>
putting Request Payments

REST, <u>PUT Bucket requestPayment</u>

R

request headers, Common Request Headers requestPayments putting, **PUT Bucket requestPayment** resources, related, Amazon S3 Resources response headers, common, Common Response Headers **REST** API, <u>REST API</u> bucket operations, **Operations on Buckets** COPY, PUT Object (Copy) DELETE, DELETE Bucket, DELETE Object GET, GET Service, GET Bucket (List Objects), GET Bucket Object versions, GET Bucket requestPayment, GET Bucket versioning, GET Object acl GET object, GET Bucket acl, GET Bucket location, GET Bucket logging, GET Object, GET Object torrent HEAD object, **HEAD Object** object operations, **Operations on Objects**

POST object, POST Object
PUT, PUT Bucket, PUT Bucket acl, PUT Bucket
logging, PUT Bucket requestPayment, PUT Bucket
versioning
PUT object, PUT Object, PUT Object acl
service operations, Operations on the Service

service REST operations, Operations on the Service SOAP operations, Operations on the Service SetBucketAccessControlPolicy SOAP, SetBucketAccessControlPolicy SetBucketLoggingStatus SOAP, SetBucketLoggingStatus SetObjectAccessControlPolicy SOAP, SetObjectAccessControlPolicy **SOAP** bucket operations, **Operations on Buckets** CopyObject, CopyObject CreateBucket, CreateBucket DeleteBucket, **DeleteBucket** DeleteObject, DeleteObject GetBucketAccessControlPolicy, **GetBucketAccessControlPolicy**

GetBucketLoggingStatus, GetBucketLoggingStatus

S

GetObjectAccessControlPolicy,
GetObjectAccessControlPolicy
GetObjectExtended, GetObjectExtended
ListAllMyBuckets, ListAllMyBuckets
ListBucket, ListBucket
object operations, Operations on Objects
PutObject, PutObject
PutObjectInline, PutObjectInline
service operations, Operations on the Service
SetBucketAccessControlPolicy,
SetBucketAccessControlPolicy
SetBucketLoggingStatus, SetBucketLoggingStatus
SetObjectAccessControlPolicy

W

WSDL, Amazon S3 API Reference Introduction