S3MultiPartProxy API

1. Initiate Multipart Upload

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

## Requests

## Syntax

POST /s3/*ObjectName?uploads* HTTP/1.1

Host: *s3proxyserver*

Date: *date*

**Response Elements**

| **Name** | **Description** |
| --- | --- |
| InitiateMultipartUploadResult | Container for response.  Type: Container  Children: Bucket, Key, UploadId  Ancestors: None |
| Bucket | Name of the bucket to which the multipart upload was initiated.  Type: string  Ancestors: InitiateMultipartUploadResult |
| Key | Object key for which the multipart upload was initiated.  Type: String  Ancestors: InitiateMultipartUploadResult |
| UploadId | ID for the initiated multipart upload.  Type: String  Ancestors: InitiateMultipartUploadResult |

## Examples

### Sample Request

This operation initiates a multipart upload for the example-object object.

POST /s3/example-object?uploads HTTP/1.1

Host: s3proxyserver

Date: Mon, 1 Nov 2010 20:34:56 GMT

### Sample Response

HTTP/1.1 200 OK

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

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

<Bucket>example-bucket</Bucket>

<Key>example-object</Key>

<UploadId>VXBsb2FkIElEIGZvciA2aWWpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZA</UploadId>

</InitiateMultipartUploadResult>

1. Upload part

This operation uploads a part in a multipart upload. You must initiate a multipart upload before you can upload any part. In response to your initiate request, service returns an upload ID, a unique identifier, that you must include in your upload part request.

## Requests

### Syntax

PUT /s3/*ObjectName*?partNumber=*PartNumber*&uploadId=*UploadId* HTTP/1.1

Host: *s3proxyserver*

Date: *date*

Content-Length: *Size*

### Request Headers

| **Name** | **Description** | **Required** |
| --- | --- | --- |
| *Content-Length* | The size of the part, in bytes. For more information, go to [http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13](http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html" \l "sec14.13" \t "_blank).  Type: Integer  Default: None | Yes |
| *Content-MD5* | The base64-encoded 128-bit MD5 digest of the part data. This header can be used as a message integrity check to verify that the part data is the same data that was originally sent. Although it is optional, we recommend using the Content-MD5 mechanism as an end-to-end integrity check. For more information, see [RFC 1864](http://www.ietf.org/rfc/rfc1864.txt" \t "_blank).  Type: String  Default: None | No |

### Response Elements

This operation does not use response elements.

### Special Errors

| **Error Code** | **Description** | **HTTP Status Code** | **SOAP Fault Code Prefix** |
| --- | --- | --- | --- |
| NoSuchUpload | The specified multipart upload does not exist. The upload ID might be invalid, or the multipart upload might have been aborted or completed. | 404 Not Found | Client |
| InternalError | We encountered an internal error. Please try again. | 500 Internal Server Error | Server |
| InvalidPart | One or more of the specified parts could not be found. The part might not have been uploaded, or the specified entity tag might not have matched the part's entity tag. | 400 Bad Request | Client |
| InvalidPartOrder | The list of parts was not in ascending order.Parts list must specified in order by part number. | 400 Bad Request | Client |
| NoSuchKey | The specified key does not exist. | 404 Not Found | Client |
| OperationAborted | A conflicting conditional operation is currently in progress against this resource. Try again. | 409 Conflict | Client |
| UnexpectedContent | This request does not support content. | 400 Bad Request | Client |

## REST Error Responses

When there is an error, the header information contains:

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

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

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

<Error>

<Code>NoSuchKey</Code>

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

<Resource>/mybucket/myfoto.jpg</Resource>

<RequestId>4442587FB7D0A2F9</RequestId>

</Error>

## Examples

### Sample Request

The following PUT request uploads a part (part number 1) in a multipart upload. The request includes the upload ID that you get in response to your Initiate Multipart Upload request.

PUT /s3/my-movie.m2ts?partNumber=1&uploadId=VCVsb2FkIElEIGZvciBlbZZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZR HTTP/1.1

Host: example-bucket.s3.amazonaws.com

Date: Mon, 1 Nov 2010 20:34:56 GMT

Content-Length: 10485760

Content-MD5: pUNXr/BjKK5G2UKvaRRrOA==

\*\*\*part data omitted\*\*\*

### Sample Response

The response includes the ETag header. You need to retain this value for use when you send the Complete Multipart Upload request.

HTTP/1.1 200 OK

Date: Mon, 1 Nov 2010 20:34:56 GMT

ETag: "b54357faf0632cce46e942fa68356b38"

Content-Length: 0

Connection: keep-alive

1. Complete Multipart Upload

This operation completes a multipart upload by assembling previously uploaded parts.

You first initiate the multipart upload and then upload all parts using the Upload Parts operation (see Upload part). After successfully uploading all relevant parts of an upload, you call this operation to complete the upload. Upon receiving this request, service concatenates all the parts in ascending order by part number to create a new object. In the Complete Multipart Upload request, you must provide the parts list. You must ensure the parts list is complete, this operation concatenates the parts you provide in the list. For each part in the list, you must provide the part number and the *ETag* header value, returned after that part was uploaded.

## Requests

### Syntax

POST /s3/*ObjectName*?uploadId=*UploadId* HTTP/1.1

Host: *s3proxyserver*

Date: *Date*

Content-Length: *Size*

<CompleteMultipartUpload>

<Part>

<PartNumber>*PartNumber*</PartNumber>

<ETag>*ETag*</ETag>

</Part>

...

</CompleteMultipartUpload>

### Request Elements

| **Name** | **Description** | **Required** |
| --- | --- | --- |
| *CompleteMultipartUpload* | Container for the request.  Ancestor: None  Type: Container  Children: One or more *Part* elements | Yes |
| *Part* | Container for elements related to a particular previously uploaded part.  Ancestor: *CompleteMultipartUpload*  Type: Container  Children: *PartNumber*, *ETag* | Yes |
| *PartNumber* | Part number that identifies the part.  Ancestor: *Part*  Type: Integer | Yes |
| *ETag* | Entity tag returned when the part was uploaded.  Ancestor: *Part*  Type: String | Yes |

### Response Elements

| **Name** | **Description** |
| --- | --- |
| CompleteMultipartUploadResult | Container for the response  Type: Container  Children: *Location*, *Bucket*, *Key*, *ETag*  Ancestors: None |
| Location | The URI that identifies the newly created object.  Type: URI  Ancestors: *CompleteMultipartUploadResult* |
| Bucket | The name of the bucket that contains the newly created object.  Type: String  Ancestors: *CompleteMultipartUploadResult* |
| Key | The object key of the newly created object.  Type: String  Ancestors: *CompleteMultipartUploadResult* |
| ETag | Entity tag that identifies the newly created object's data. Objects with different object data will have different entity tags. The entity tag is an opaque string. The entity tag may or may not be an MD5 digest of the object data. If the entity tag is not an MD5 digest of the object data, it will contain one or more nonhexadecimal characters and/or will consist of less than 32 or more than 32 hexadecimal digits.  Type: String  Ancestors: *CompleteMultipartUploadResult* |

### Special Errors

See Upload Part.

## Examples

### Sample Request

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

POST /s3/example-object?uploadId=AAAsb2FkIElEIGZvciBlbHZpbmcncyWeeS1tb3ZpZS5tMnRzIRRwbG9hZA HTTP/1.1

Host: s3proxyserver

Date: Mon, 1 Nov 2010 20:34:56 GMT

Content-Length: 391

<CompleteMultipartUpload>

<Part>

<PartNumber>1</PartNumber>

<ETag>"a54357aff0632cce46d942af68356b38"</ETag>

</Part>

<Part>

<PartNumber>2</PartNumber>

<ETag>"0c78aef83f66abc1fa1e8477f296d394"</ETag>

</Part>

<Part>

<PartNumber>3</PartNumber>

<ETag>"acbd18db4cc2f85cedef654fccc4a4d8"</ETag>

</Part>

</CompleteMultipartUpload>

### Sample Response

The following response indicates that an object was successfully assembled.

HTTP/1.1 200 OK

Date: Mon, 1 Nov 2010 20:34:56 GMT

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

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

<Location>http://Example-Bucket.s3.amazonaws.com/Example-Object</Location>

<Bucket>Example-Bucket</Bucket>

<Key>Example-Object</Key>

<ETag>"3858f62230ac3c915f300c664312c11f-9"</ETag>

</CompleteMultipartUploadResult>

1. Abort Multipart Upload

This operation aborts a multipart upload. After a multipart upload is aborted, no additional parts can be uploaded using that upload ID.

## Requests

### Syntax

DELETE /s3/*ObjectName*?uploadId=*UploadId* HTTP/1.1

Host: *s3proxyserver*

Date: *Date*

## Examples

### Sample Request

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

DELETE /s3/example-object?uploadId=VXBsb2FkIElEIGZvciBlbHZpbmcncyBteS1tb3ZpZS5tMnRzIHVwbG9hZ HTTP/1.1

Host: s3proxyserver

Date: Mon, 1 Nov 2010 20:34:56 GMT

### Sample Response

HTTP/1.1 204 OK

Date: Mon, 1 Nov 2010 20:34:56 GMT

Content-Length: 0

Connection: keep-alive

1. Check Multipart Upload

This operation returns the URL and the status of the overall upload

## Requests

### Syntax

GET /*check*?uploadId=*UploadId* HTTP/1.1

Host: *s3proxyserver*

Date: *Date*

### Sample Response

HTTP/1.1 200 OK

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

<CheckMultipartUploadResult>

<Location>http://Example-Bucket.s3.amazonaws.com/Example-Object</Location>

<StatusCode>3</StatusCode>

</CheckMultipartUploadResult>

Status code:

0 – init, 1 – uploaded, 2- combined, 3- read, 4 – aborted, 5 –error