This page discusses Cloud Storage resumable uploads in JSON API and XML API. To learn how to perform a resumable upload, see Performing Resumable Uploads (/storage/docs/performing-resumable-uploads). Because you don't have to restart large file uploads from the beginning, resumable uploads can reduce your bandwidth usage when there is a network failure. A completed resumable upload is considered one Class A operation (/storage/pricing#operations-pricing).

Cloud Storage provides a resumable data transfer feature that lets you resume upload operations after a communication failure has interrupted the flow of data. Resumable uploads are useful if you are transferring large files, because the likelihood of a network interruption or some other transmission failure is high. By using the resumable upload feature, you can reduce your bandwidth usage (and therefore your bandwidth cost) because you do not have to restart large file uploads from the beginning. For tips on uploading to Cloud Storage, see best practices (/storage/docs/best-practices).

The Cloud Storage JSON and XML APIs provide two standard HTTP methods for uploading data: POST Object (/storage/docs/xml-api/post-object) and PUT Object (/storage/docs/xml-api/put-object). To implement a resumable upload, you use both of these methods in conjunction with various headers and query string parameters.

If you are sending small files over a reliable network connection, you can use a simple upload (/storage/docs/uploading-objects).

ant: When you initiate a resumable upload session, a session URI is returned that you use for creating and resuming
ls. This session URI acts as an authentication token, so the upload PUT requests don't need to be signed. Be judicious
g the session URI and only transmit it over HTTPS, because it can be used by anyone to upload data to the target bu
ıt any further authentication.

When you upload objects with the JSON API, you use a special URI that differs from the URI you use for most JSON API requests, including requests to upload object metadata:

- **For object uploads, use the `/upload` URI.** The format of the `/upload` endpoint is the standard resource URI with an `/upload` prefix. Use this URI when transferring the object data itself.

  For example, for a bucket named `myBucket`:

- **For other requests, use the standard resource URI.** This includes adding or updating metadata values (/storage/docs/viewing-editing-metadata#edit) for an existing object.

  For example, for a metadata patch to an object named `myObject` in a bucket named `myBucket`:

You can include a `Content-Type` (/storage/docs/xml-api/reference-headers#contenttype) request header if you want to specify a content type for the file you are uploading. If you do not specify a content type, the Cloud Storage system sets the content type to `application/octet-stream` (https://tools.ietf.org/html/rfc2046#section-4.5.1) when it serves the object you are uploading.

The `x-goog-resumable` header is a Cloud Storage extension (custom) header. The header notifies the Cloud Storage system that you want to initiate a resumable upload. The header can be used only with a POST Object request and can be used only for resumable uploads.

In addition, you must use the standard Cloud Storage host name in the request, and you must authenticate the POST Object request just as you would any authenticated request. For more information, see Request Endpoints (/storage/docs/request-endpoints#typical) and Authentication (/storage/docs/authentication).

The resumable upload mechanism supports transfers where the file size is not known in advance. This can be useful for cases like compressing an object on-the-fly while uploading, since it's difficult to predict the exact file size for the compressed file at the start of a transfer. The mechanism is useful

either if you want to stream a transfer that can be resumed after being interrupted, or if chunked transfer encoding does not work for your application.

- A session URI expires after one week. We recommend that you start a resumable upload as soon as you obtain the session URI and that you resume an interrupted upload shortly after the interruption occurred.
- If you use an expired session URI in a request, you receive a `400 Bad Request` status code. In this case, you will have to initiate a new resumable upload, obtain a new session URI, and start the upload from the beginning using the new session URI.
- When you initiate a resumable upload session, a session URI is returned that you use for creating and resuming uploads. This session URI acts as an authentication token, so the upload `PUT` requests don't need to be signed. Be judicious in sharing the session URI and only transmit it over HTTPS, because it can be used by anyone to upload data to the target bucket without any further authentication.
- Also, you should retry any requests that return the following status codes:
  - `408 Request Timeout`
  - `500 Internal Server Error`
  - `502 Bad Gateway`
  - `503 Service Unavailable`
  - `504 Gateway Timeout`
- When performing a resumable upload, handle `404 Not Found` errors by starting the entire upload over from the beginning.

When performing retry requests, use truncated exponential backoff (/storage/docs/exponential-backoff).

- In addition, we recommend that you request an integrity check of the final uploaded object to be sure that it matches the source file. You can do this by calculating the MD5 digest of the source file and adding it to the `Content-MD5` (/storage/docs/xml-api/reference-headers#contentmd5) request header. Checking the integrity of the uploaded file is particularly important if you are uploading a large file over a long period of time, because there is an increased likelihood of the source file being modified over the course of the upload operation.

- Resumable uploads are pinned in the region they start in. For example, if you create a resumable upload URL in the US and give it to a client in Asia, the upload will still go through the US. Continuing a resumable upload in a region where it wasn't initiated can cause slow uploads.
- If you use Compute Engine instances with processes that POST to Cloud Storage to initiate a resumable upload, then you should use Compute Engine instances in the same locations as your Cloud Storage buckets. You can then use a geo IP service to pick the Compute Engine region to which you route customer requests, which will help keep traffic localized to a geo-region.

If you receive a `308 Resume Incomplete` response with no `Range` header, it's possible some bytes have been received by Cloud Storage but were not yet persisted at the time Cloud Storage received the query. Retransmitting from the beginning of the file in this case is somewhat wasteful. To reduce the likelihood of this case, you can wait a few seconds after the first `308` response and then query a second time; at that point you might receive a `Range` header, allowing you to avoid retransmitting the start of the file. If you don't receive a `Range` header on this second try you should not continue to wait and try a third time, as continuing to re-query could lead to a "hung" upload in the case where Cloud Storage truly has not received any data for the upload.