Storage Products (https://cloud.google.com/products/storage/)
Documentation (https://cloud.google.com/storage/docs/) Guides

Canonical requests

Canonical requests define the elements of a request that a user must include when sending V4 signature-authenticated requests, such as signed URLs (https://cloud.google.com/storage/docs/access-control/signed-urls), to Cloud Storage. If you are sending V2 signed URLs, see the V2 signing process (https://cloud.google.com/storage/docs/access-control/signed-urls-v2)

Overview

A *canonical request* is a string that represents a specific HTTP request to Cloud Storage. You use a canonical request along with a cryptographic key, such as an RSA key, to create a *signature* that is then included in the actual request as authentication.

A canonical request includes information such as the HTTP verb, query string parameters, and headers expected to be used in the actual request, as well as the object, bucket, or other resource to be requested.

A canonical request ensures that when Cloud Storage receives the request, it can calculate the same signature that you calculated. If your version and the version calculated by Cloud Storage don't match, the request fails.

Structure

Canonical requests have the following structure, including the use of newlines between each element:

```
HTTP_VERB
PATH_TO_RESOURCE
CANONICAL_QUERY_STRING
CANONICAL_HEADERS

SIGNED_HEADERS
PAYLOAD
```

HTTP verbs

Signed requests can use the following HTTP verbs, which must be specified as part of the canonical request:

- `DELETE`
- `GET`
- `HEAD`
- `POST`[1]
- `PUT`

[1] Signed URLs do not support `POST` requests, except when working with resumable uploads (https://cloud.google.com/storage/docs/access-control/signed-urls#signing-resumable).

Resource path

Canonical requests include the path to the resource that the request applies to. The path to the resource is everything that follows the host name but precedes any query string.

For example, if the Cloud Storage URL is `https://storage.googleapis.com/example-bucket/cat-pics/tabby.jpeg`, then the path to the resource is `/example-bucket/cat-pics/tabby.jpeg`.

If you use an alternative Cloud Storage URL such as `https://example-bucket.storage.googleapis.com/cat-pics/tabby.jpeg` then the path to the resource is `/cat-pics/tabby.jpeg`.

For additional URL endpoints that can be used with signed URLs, see XML API request endpoints (https://cloud.google.com/storage/docs/request-endpoints).

When defining the resource path, you must percent encode (https://en.wikipedia.org/wiki/Percent-encoding) the following reserved characters: `? =!#$&'()*+,:;@[]."` Any other percent encoding used in the URL should also be included in the resource path.

Canonical Query string

Canonical requests include any query string parameters that *must* be subsequently included in signed requests that use the relevant signature. However, such signed requests *may* include additional query string parameters that were not specified in the canonical request. The query string specified in the canonical request is called the *canonical query string*

The query string is everything that follows the question mark (`?`) at the end of the resource path.

For example, if the Cloud Storage URL is `https://storage.googleapis.com/example-bucket/cat-pics/tabby.jpeg?generation=1360887697105000&userProject=my-project`, then the query string is `generation=1360887697105000&userProject=my-project`.

When constructing the canonical query string:

- The parameters in the query string must be sorted by name using a lexicographical sort by code point value.
- Each parameter in the query string must be separated with `&`.
- If your canonical query string is empty, this portion of the overall canonical request is simply a new line (`/n`).

Required query string parameters

Most query string parameters are added as needed, but the following *must* be included in your canonical request when you intend to use it to make a signed URL (https://cloud.google.com/storage/docs/access-control/signed-urls):

- `X-Goog-Algorithm`: The algorithm you will use to sign the URL. Valid values are `GOOG4-RSA-SHA256` and `GOOG4-HMAC-SHA256`.
- `X-Goog-Credential`: The credentials you will use to sign the URL. Credentials consist of an authorizer and a credential scope (https://cloud.google.com/storage/docs/access-control/signed-urls#credential-scope) given in the format: `[AUTHORIZER]%2F[CREDENTIAL_SCOPE]`. The authorizer can be a service account name (https://cloud.google.com/iam/docs/service-accounts) or an HMAC access key.
- `X-Goog-Date`: The current date and time, in the ISO 8601 (https://en.wikipedia.org/wiki/ISO_8601) basic format `YYYYMMDD'T'HHMMSS'Z'`.
- `X-Goog-Expires`: The lifetime of the signed URL, measured in seconds from `X-Goog-Date`. The longest expiration value is 604800 seconds (7 days).

- `X-Goog-SignedHeaders`: A semicolon-separated list of names of headers defined in the canonical request. These are also known as *signed headers*. `host` must be one of the header names.

These query string parameters subsequently must be used in the signed URL itself, along with the `X-Goog-Signature` query string parameter, which contains the signature authenticating the request.

Note: When you use an HMAC key to sign your canonical request, Cloud Storage also accepts these query string parameters if they are consistently prefixed with `X-Amz-` instead of `X-Goog-`.

Canonical Headers

Canonical requests include any headers that *must* be subsequently included in signed requests that use the relevant signature. However, such signed requests *may* include additional headers that were not specified in the canonical request, except as noted in required headers (#required-headers). Headers specified in the canonical request are called *canonical headers*

Canonical headers can include custom headers as well as extension headers (https://cloud.google.com/storage/docs/xml-api/reference-headers) that begin with `x-goog`.

When specifying canonical headers, keep in mind the following:

- Make all header names lowercase.
- Sort all headers by header name using a lexicographical sort by code point value.
- Separate each header with a newline (`/n`).
- Eliminate duplicate header names by creating one header name with a comma-separated list of values. Be sure there is no whitespace between the values, and be sure that the order of the comma-separated list matches the order that the headers appear in your request. For more information, see RFC 7230 section 3.2 (https://tools.ietf.org/html/rfc7230#section-3.2).
- Replace any folding whitespace or newlines (CRLF or LF) with a single space. For more information about folding whitespace, see RFC 7230, section 3.2.4. (https://tools.ietf.org/html/rfc7230#section-3.2.4).
- Remove any whitespace around the colon that appears after the header name.

For example, using the custom header `x-goog-acl: private` without removing the space after the colon returns a `403 Forbidden` error, because the request signature you calculate does not match the signature Google calculates.

Example

If you have the following set of headers:

```
host: storage.googleapis.com
content-type: text/plain
x-goog-meta-reviewer: jane
x-goog-meta-reviewer: john
```

The construction of the canonical headers in the canonical request would be:

```
content-type:text/plain
host:storage.googleapis.com
x-goog-meta-reviewer:jane,john
```

Note: In the above examples, newlines are shown as actual new lines and not `\n`.

Required canonical headers

Most headers, such as `content-type` and extension headers, are added as needed, but the following header is required in every signed request:

- `host`: The URI used to access Cloud Storage.

Additionally, the following headers cannot be used in requests that use the signature *unless* they are explicitly defined as canonical headers.

- `x-goog-project-id`
- `x-goog-copy-source`
- `x-goog-metadata-directive`
- `x-amz-copy-source`
- `x-amz-metadata-directive`

Signed headers

A *signed header* is the name portion of a canonical header.

To create the signed headers list, convert all header names to lowercase, sort them by character code, and use a semicolon (`;`) to separate each.

Example

If you have the following set of headers:

```
host: storage.googleapis.com
content-type: text/plain
x-goog-meta-reviewer: jane
x-goog-meta-reviewer: john
```

The construction of the signed headers in the canonical request would be:

```
content-type;host;x-goog-meta-reviewer
```

Payload

- If your canonical request will be used to create a signed URL (https://cloud.google.com/storage/docs/access-control/signed-urls), this value should be simply the string `UNSIGNED-PAYLOAD`.
- If your canonical request will be used as part of a request that uses an `Authorization` header, this value should be a hex-encoded, SHA-256 hashed request payload. If the payload is empty, use an empty string as the input to the hash function. An example of a hashed payload (in this case an empty payload) is:

  ```
  e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
  ```

Example

The following is an example of a properly formed canonical request, with newlines shown as actual new lines and not `\n`:

```
GET
/example-bucket/tabby.jpeg

host:storage.googleapis.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b85
x-amz-date:20190301T190859Z

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

What's next

- Build a request that uses a canonical request (https://cloud.google.com/storage/docs/migrating#authentication).
- Build signed URLs (https://cloud.google.com/storage/docs/access-control/signing-urls-manually), which use canonical requests.
- Learn more about signed URLs (https://cloud.google.com/storage/docs/access-control/signed-urls).

Last updated January 8, 2020.