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

Object metadata

This page discusses the metadata fields that are stored along with objects in Cloud Storage. To learn how to view and set metadata on objects, see Viewing and Editing Object Metadata (https://cloud.google.com/storage/docs/viewing-editing-metadata).

Introduction

Objects stored in Cloud Storage have *metadata* associated with them. Metadata identifies properties of the object, as well as specifies how the object should be handled when it's accessed. Metadata exists as *key:value pairs*. For example, the storage class (https://cloud.google.com/storage/docs/storage-classes) of an object is represented by the metadata entry `storageClass:STANDARD`. `storageClass` is the *key* for the metadata, and all objects have such a key associated with them. `STANDARD` specifies the *value* this specific object has, and the value varies from object to object.

The mutability of metadata varies: some metadata you can edit at any time, some metadata you can only set at the time the object is created, and some metadata you can only view. For example, you can edit the value of the `Cache-Control` metadata at any time, but you can only assign the `storageClass` metadata when the object is created or rewritten, and you cannot directly edit the value for the `generation` metadata, though the `generation` value changes when the object is overwritten.

Editable metadata

There are two categories of metadata that users can change (https://cloud.google.com/storage/docs/viewing-editing-metadata) for objects:

- **Fixed-key metadata**: Metadata whose keys are set, but for which you can specify a value.
- **Custom metadata**: Metadata that you add by specifying both a key and a value associated with the key.

When editing metadata, you should generally avoid non-ascii characters, because they are not permitted in HTTP headers, which the XML API uses. When using the XML API, there is also a 16 KB limit to the combined size of the request URL and HTTP headers, so the total size of your metadata should take this limit into account.

Fixed-key metadata

You can edit the following metadata for objects, though you must have sufficient permission (https://cloud.google.com/storage/docs/access-control/iam) to do so:

- Access control metadata (#access-control)
- Cache-Control (#cache-control)
- Content-Disposition (#content-disposition)
- Content-Encoding (#content-encoding)
- Content-Language (#content-language)
- Content-Type (#content-type)

Access control metadata

Cloud Storage uses Identity and Access Management (IAM) (https://cloud.google.com/storage/docs/access-control/iam) and Access Control Lists (ACLs) (https://cloud.google.com/storage/docs/access-control/lists) to control access to objects. Use these links to learn about these access control methods and associated metadata.

Cache-Control

The `Cache-Control` metadata can specify two different aspects of how data is served from Cloud Storage: whether the data can be cached and whether the data can be transformed.

Note: `Cache-Control` is also a header you can specify in your HTTP requests for an object; however, Cloud Storage ignores this header and sets response `Cache-Control` headers based on the stored metadata values.

Caching data

The `Cache-Control` metadata allows you to control whether and for how long browser and Internet caches are allowed to cache your objects, which can then be served to satisfy future requests. `Cache-Control` only applies when accessing objects that:

- Are publicly accessible (https://cloud.google.com/storage/docs/access-control/making-data-public).
- Are not stored in a bucket that has Requester Pays (https://cloud.google.com/storage/docs/requester-pays) enabled.

Setting the value of `Cache-Control` to `public` means the object may be cached anywhere. Setting the value to `private` means the object may be cached in a requester's local cache. Setting the value to `no-cache` means the object may be cached, but cannot be used to satisfy future requests unless first validated by Cloud Storage.

You can also add a value of `max-age=[TIME_IN_SECONDS]` to the `Cache-Control` metadata. The `max-age` value indicates the length of time an object may be cached before it's considered stale. Stale objects are not served from caches, except in special circumstances (https://tools.ietf.org/html/rfc7234#section-4.2.4).

If an applicable object doesn't have a `Cache-Control` metadata entry, Cloud Storage uses the default value of:

- `public, max-age=3600`.

★ **Important:** Currently, if the object is stored in a bucket that has uniform bucket-level access (https://cloud.google.com/storage/docs/uniform-bucket-level-access) enabled, responses serving the object use a default value of `private`. This behavior is temporary.

If you allow caching, downloads may continue to receive older versions of an object, even after uploading a newer version. This is because the older version remains "fresh" in the cache for a period of time determined by `max-age`. Additionally, because objects can be cached at various places on the Internet, there is no way to force a cached object to expire globally. If you want to prevent serving cached versions of publicly readable objects, set `Cache-Control:no-cache, max-age=0` on the object.

Transforming data

`Cache-Control` metadata also allows you to serve objects as they are stored, without applying any transformations (https://cloud.google.com/storage/docs/transcoding) to the data, such as removing gzip content-encoding for incompatible clients. To serve an object as-is, set `Cache-Control:no-transform`.

Content-Disposition

The `Content-Disposition` metadata specifies presentation information about the data being transmitted. Setting `Content-Disposition` allows you to control presentation style of the content, for example determining whether an attachment should be automatically displayed or whether some form of action from the user should be required to open it. See https://tools.ietf.org/html/rfc6266 (https://tools.ietf.org/html/rfc6266) for the `Content-Disposition` specification.

Content-Encoding

The `Content-Encoding` metadata can be used to indicate that an object is compressed, while still maintaining the object's underlying `Content-Type`. For example, a text file that is gzip compressed can have the fact that it's a text file indicated in `Content-Type` and the fact that it's gzip compressed indicated in `Content-Encoding`. You should ensure that files are, in fact, compressed using the specified `Content-Encoding` before uploading them, or else unexpected behavior can occur when attempting to download the objects. For more information, see the Transcoding page (https://cloud.google.com/storage/docs/transcoding).

For compressible content, such as text, using `Content-Encoding: gzip` saves network and storage costs and improves content serving performance. However, for content that is already inherently compressed, such as archives and many media formats, applying another level of compression and marking it in the `Content-Encoding` metadata is typically detrimental to both object size and performance and should be avoided.

Content-Language

The `Content-Language` metadata indicates the language(s) that the object is intended for. Refer to ISO 639-1 (https://www.loc.gov/standards/iso639-2/php/code_list.php) language codes for the supported values of this metadata.

Content-Type

The most commonly set metadata is `Content-Type` (also known as media type), which allows browsers to render the object properly. All objects have a value specified in their `Content-Type` metadata, but this value does not have to match the underlying type of the object. For example, if the `Content-Type` is not specified by the uploader and cannot be determined, it is set to `application/octet-stream` or `application/x-www-form-urlencoded`, depending on how you uploaded the object. Refer to the IANA Media Types (https://www.iana.org/assignments/media-types/media-types.xhtml) page for a list of valid content types.

Custom metadata

Custom metadata is metadata that you can add and remove. To create custom metadata, you specify both a value and a key. Once you have created a custom metadata `key:value` pair, you can delete the key or change the value.

The Viewing and Editing Metadata page (https://cloud.google.com/storage/docs/viewing-editing-metadata) includes information on setting custom metadata. Note that using custom metadata incurs storage and network costs (https://cloud.google.com/storage/pricing#storage-notes).

Non-editable metadata

Some metadata cannot be edited directly. This metadata is set at the time of object creation or rewrite. As part of the object creation or rewrite, you can set some such metadata, such as the storage class (https://cloud.google.com/storage/docs/storage-classes) of the object or customer-managed encryption keys (https://cloud.google.com/storage/docs/encryption/customer-managed-keys). Other metadata is automatically added and can only be viewed, such the generation number (https://cloud.google.com/storage/docs/key-terms#generation-numbers) of the object or the time of creation.

What's next

- View and edit object metadata (https://cloud.google.com/storage/docs/viewing-editing-metadata).

- Learn about available storage classes (https://cloud.google.com/storage/docs/storage-classes).
- Learn about the generation number associated with an object (https://cloud.google.com/storage/docs/generations-preconditions).
- Learn about object holds (https://cloud.google.com/storage/docs/bucket-lock#object-holds).

Last updated December 25, 2019.