## Google Photos

The rclone backend for Google Photos is a specialized backend for transferring photos and videos to and from Google Photos.

**NB** The Google Photos API which rclone uses has quite a few limitations, so please read the limitations section carefully to make sure it is suitable for your use.

## Configuring Google Photos

The initial setup for google cloud storage involves getting a token from Google Photos which you need to do in your browser. `rclone config` walks you through it.

Here is an example of how to make a remote called `remote`. First run:

```
rclone config
```

This will guide you through an interactive setup process:

```
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> remote
Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
[snip]
XX / Google Photos
   \ "google photos"
[snip]
Storage> google photos
** See help for google photos backend at: https://rclone.org/googlephotos/ **

Google Application Client Id
Leave blank normally.
Enter a string value. Press Enter for the default ("").
client_id>
Google Application Client Secret
Leave blank normally.
Enter a string value. Press Enter for the default ("").
client_secret>
Set to make the Google Photos backend read only.

If you choose read only then rclone will only request read only access
to your photos, otherwise rclone will request full access.
Enter a boolean value (true or false). Press Enter for the default ("false").
read_only>
Edit advanced config? (y/n)
y) Yes
n) No
y/n> n
Remote config
Use auto config?
 * Say Y if not sure
 * Say N if you are working on a remote or headless machine
y) Yes
n) No
y/n> y
If your browser doesn't open automatically go to the following link: http://127.0.0.1:53682/a
Log in and authorize rclone for access
Waiting for code...
Got code

*** IMPORTANT: All media items uploaded to Google Photos with rclone
*** are stored in full resolution at original quality.  These uploads
*** will count towards storage in your Google Account.

--------------------
```

```
[remote]
type = google photos
token = {"access_token":"XXX","token_type":"Bearer","refresh_token":"XXX","expiry":"2019-06-2
--------------------
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
```

Note that rclone runs a webserver on your local machine to collect the token as returned from Google if you use auto config mode. This only runs from the moment it opens your browser to the moment you get back the verification code. This is on `http://127.0.0.1:53682/`and this may require you to unblock it temporarily if you are running a host firewall, or use manual mode.

This remote is called `remote` and can now be used like this

See all the albums in your photos

```
rclone lsd remote:album
```

Make a new album

```
rclone mkdir remote:album/newAlbum
```

List the contents of an album

```
rclone ls remote:album/newAlbum
```

Sync `/home/local/images` to the Google Photos, removing any excess files in the album.

```
rclone sync -i /home/local/image remote:album/newAlbum
```

## Layout

As Google Photos is not a general purpose cloud storage system the backend is laid out to help you navigate it.

The directories under `media` show different ways of categorizing the media. Each file will appear multiple times. So if you want to make a backup of your google photos you might choose to backup `remote:media/by-month`. (**NB** `remote:media/by-day` is rather slow at the moment so avoid for syncing.)

Note that all your photos and videos will appear somewhere under `media`, but they may not appear under `album` unless you've put them into albums.

```
/
- upload
    - file1.jpg
    - file2.jpg
    - ...
- media
    - all
        - file1.jpg
        - file2.jpg
        - ...
    - by-year
        - 2000
            - file1.jpg
            - ...
        - 2001
            - file2.jpg
            - ...
        - ...
    - by-month
        - 2000
            - 2000-01
                - file1.jpg
                - ...
            - 2000-02
                - file2.jpg
                - ...
        - ...
    - by-day
        - 2000
            - 2000-01-01
                - file1.jpg
                - ...
            - 2000-01-02
                - file2.jpg
                - ...
        - ...
- album
    - album name
    - album name/sub
- shared-album
    - album name
    - album name/sub
- feature
    - favorites
        - file1.jpg
        - file2.jpg
```

There are two writable parts of the tree, the `upload` directory and sub directories of the `album` directory.

The `upload` directory is for uploading files you don't want to put into albums. This will be empty to start with and will contain the files you've uploaded for one rclone session only, becoming empty again when you restart rclone. The use case for this would be if you have a load of files you just want to once off dump into Google Photos. For repeated syncing, uploading to `album` will work better.

Directories within the `album` directory are also writeable and you may create new directories (albums) under `album`. If you copy files with a directory hierarchy in there then rclone will create albums with the `/` character in them. For example if you do

```
rclone copy /path/to/images remote:album/images
```

and the images directory contains

```
images
    - file1.jpg
    dir
        file2.jpg
    dir2
        dir3
            file3.jpg
```

Then rclone will create the following albums with the following files in

- images
    - file1.jpg
- images/dir
    - file2.jpg
- images/dir2/dir3
    - file3.jpg

This means that you can use the `album` path pretty much like a normal filesystem and it is a good target for repeated syncing.

The `shared-album` directory shows albums shared with you or by you. This is similar to the Sharing tab in the Google Photos web interface.

## Limitations

Only images and videos can be uploaded. If you attempt to upload non videos or images or formats that Google Photos doesn't understand, rclone will upload the file, then Google Photos will give an error when it is put turned into a media item.

Note that all media items uploaded to Google Photos through the API are stored in full resolution at "original quality" and **will** count towards your storage quota in your Google Account. The API does **not** offer a way to upload in "high quality" mode..

`rclone about` is not supported by the Google Photos backend. Backends without this capability cannot determine free space for an rclone mount or use policy `mfs` (most free space) as a member of an rclone union remote.

See List of backends that do not support rclone about See rclone about

### Downloading Images

When Images are downloaded this strips EXIF location (according to the docs and my tests). This is a limitation of the Google Photos API and is covered by bug #112096115.

**The current google API does not allow photos to be downloaded at original resolution. This is very important if you are, for example, relying on "Google Photos" as a backup of your photos. You will not be able to use rclone to redownload original images. You could use 'google takeout' to recover the original photos as a last resort**

### Downloading Videos

When videos are downloaded they are downloaded in a really compressed version of the video compared to downloading it via the Google Photos web interface. This is covered by bug #113672044.

### Duplicates

If a file name is duplicated in a directory then rclone will add the file ID into its name. So two files called `file.jpg` would then appear as `file {123456}.jpg` and `file {ABCDEF}.jpg` (the actual IDs are a lot longer alas!).

If you upload the same image (with the same binary data) twice then Google Photos will deduplicate it. However it will retain the filename from the first upload which may confuse rclone. For example if you uploaded an image to `upload` then uploaded the same image to `album/my_album` the filename of the image in `album/my_album` will be what it was uploaded with initially, not what you uploaded it with to `album`. In practise this shouldn't cause too many problems.

### Modified time

The date shown of media in Google Photos is the creation date as determined by the EXIF information, or the upload date if that is not known.

This is not changeable by rclone and is not the modification date of the media on local disk. This means that rclone cannot use the dates from Google Photos for syncing purposes.

### Size

The Google Photos API does not return the size of media. This means that when syncing to Google Photos, rclone can only do a file existence check.

It is possible to read the size of the media, but this needs an extra HTTP HEAD request per media item so is **very slow** and uses up a lot of transactions. This can be enabled with the `--gphotos-read-size` option or the `read_size = true` config parameter.

If you want to use the backend with `rclone mount` you may need to enable this flag (depending on your OS and application using the photos) otherwise you may not be able to read media off the mount. You'll need to experiment to see if it works for you without the flag.

## Albums

Rclone can only upload files to albums it created. This is a limitation of the Google Photos API.

Rclone can remove files it uploaded from albums it created only.

## Deleting files

Rclone can remove files from albums it created, but note that the Google Photos API does not allow media to be deleted permanently so this media will still remain. See bug #109759781.

Rclone cannot delete files anywhere except under `album`.

## Deleting albums

The Google Photos API does not support deleting albums - see bug #135714733.

## Standard Options

Here are the standard options specific to google photos (Google Photos).

### --gphotos-client-id

OAuth Client Id Leave blank normally.

- Config: client_id
- Env Var: RCLONE_GPHOTOS_CLIENT_ID
- Type: string
- Default: ""

### --gphotos-client-secret

OAuth Client Secret Leave blank normally.

- Config: client_secret
- Env Var: RCLONE_GPHOTOS_CLIENT_SECRET
- Type: string
- Default: ""

### --gphotos-read-only

Set to make the Google Photos backend read only.

If you choose read only then rclone will only request read only access to your photos, otherwise rclone will request full access.

- Config: read_only

- Env Var: RCLONE_GPHOTOS_READ_ONLY
- Type: bool
- Default: false

### Advanced Options

Here are the advanced options specific to google photos (Google Photos).

#### --gphotos-token

OAuth Access Token as a JSON blob.

- Config: token
- Env Var: RCLONE_GPHOTOS_TOKEN
- Type: string
- Default: ""

#### --gphotos-auth-url

Auth server URL. Leave blank to use the provider defaults.

- Config: auth_url
- Env Var: RCLONE_GPHOTOS_AUTH_URL
- Type: string
- Default: ""

#### --gphotos-token-url

Token server url. Leave blank to use the provider defaults.

- Config: token_url
- Env Var: RCLONE_GPHOTOS_TOKEN_URL
- Type: string
- Default: ""

#### --gphotos-read-size

Set to read the size of media items.

Normally rclone does not read the size of media items since this takes another transaction. This isn't necessary for syncing. However rclone mount needs to know the size of files in advance of reading them, so setting this flag when using rclone mount is recommended if you want to read the media.

- Config: read_size
- Env Var: RCLONE_GPHOTOS_READ_SIZE
- Type: bool
- Default: false

### --gphotos-start-year

Year limits the photos to be downloaded to those which are uploaded after the given year

- Config: start_year
- Env Var: RCLONE_GPHOTOS_START_YEAR
- Type: int
- Default: 2000

### --gphotos-include-archived

Also view and download archived media.

By default rclone does not request archived media. Thus, when syncing, archived media is not visible in directory listings or transferred.

Note that media in albums is always visible and synced, no matter their archive status.

With this flag, archived media are always visible in directory listings and transferred.

Without this flag, archived media will not be visible in directory listings and won't be transferred.

- Config: include_archived
- Env Var: RCLONE_GPHOTOS_INCLUDE_ARCHIVED
- Type: bool
- Default: false