

Forum:[Hack Forums](#)

Thread: [Turn Massive MEGA Files Into Simple HTTP Downloads Quota Bypass, Unlimited Download](#)

sundaran - September 19, 2026, 10:04 pm

MEGA INDEX CLOUDFLARE

The project is available on GitHub:
github.com/developeranaz/MEGA-INDEX-CLOUDFLARE

? INTRODUCTION

I recently tested MEGA INDEX CLOUDFLARE, an open-source project that provides another way to access files stored on MEGA through a Cloudflare Worker.

Instead of relying entirely on the normal MEGA interface, it processes a MEGA link and presents the available files through a generated index.

This post summarizes my testing experience and the main features I found interesting.

? HOW IT WORKS

Code MEGA LINK

?

CLOUDFLARE WORKER

?

FILE INDEX

?

BROWSER / DOWNLOAD SOFTWARE

The Worker acts as an intermediate layer between the MEGA share and the user, providing a browsable file-listing interface.

? TWO MODES

1. ? SHARED LINK MODE

This is the simpler mode.

Code MEGA SHARED LINK

?

CLOUDFLARE WORKER

?

FILE INDEX

?

SELECT FILE

?

DOWNLOAD

A public/shared MEGA link can be processed without requiring a MEGA account login.

2. ? ACCOUNT MODE

This mode is intended for working with a user's own MEGA account and provides functionality beyond public links.

Because authentication information is involved, users should understand how credentials are handled before using a publicly hosted Worker.

Since the project is open source, users can inspect the source and deploy their own instance.

? TESTING EXPERIENCE

The main feature I tested was converting a MEGA shared link into a browsable file index.

Instead of navigating the normal MEGA interface, folders and files are presented in a familiar file-listing style.

Code Open MEGA link

?

Generate index

?

Browse folders

?

Select file

?

Access / download

This is particularly convenient when a shared folder contains many files and only specific files are required.

? FOLDER BROWSING

Nested folders can be browsed through the generated index:

Code ROOT

+-- FOLDER 1

| +-- FILE A

| +-- FILE B

+-- FOLDER 2

```
| +-- FILE C
| +-- FILE D
+-- FOLDER 3
+-- FILE E
```

This makes locating individual files easier without dealing with the entire folder at once.

? HTTP-BASED ACCESS

Files exposed through the generated interface can potentially be accessed by software supporting normal HTTP URLs.

Depending on the software and file, this can include browsers, download managers and command-line tools.

This provides a more familiar HTTP-oriented workflow for accessing MEGA-hosted content.

? LARGE FILES

A particularly interesting use case is large files.

For example, a 100 GB file may take many hours to download, and interruptions can make the process inconvenient depending on the software.

The project provides another access path:

Code MEGA

?

LARGE FILE

?

CLOUDFLARE WORKER

?

HTTP ACCESS

?

BROWSER / DOWNLOAD SOFTWARE

The purpose is not to magically increase the file's speed or reduce its size, but to provide another way of accessing the MEGA-hosted file.

? CURRENT LIMITATION

The main limitation becomes noticeable with large folder structures containing hundreds or thousands of files.

Downloading an entire complicated folder structure may require manually navigating and downloading files:

For small folders this is not a major problem, but larger collections could benefit from better bulk-download functionality.

? EXPERIMENTAL ZIP STREAMING

The project also has an interesting experimental concept: streaming a MEGA folder directly as a ZIP archive.

Code MEGA FILES

?

FETCH DATA

?

CLOUDFLARE WORKER

?

BUILD ZIP STREAM

?

USER DEVICE

The idea is to progressively fetch files, add them to a ZIP stream and send the resulting data to the user instead of first storing the entire archive on a server.

However, this feature should currently be considered EXPERIMENTAL / WORK IN PROGRESS.

Streaming MEGA data while simultaneously creating a ZIP archive introduces additional data-handling and streaming challenges.

The normal file-index functionality and ZIP streaming should therefore be considered separately.

? WHY CLOUDFLARE?

The project uses Cloudflare Workers, allowing users to deploy the Worker through their own Cloudflare account.

Code YOUR CLOUDFLARE ACCOUNT

?

CLOUDFLARE WORKER

?

MEGA PROCESSING

?

FILE INDEX

Self-hosting provides more control over the deployment and is useful for technically experienced users.

? OPEN SOURCE & SECURITY

Users can:

- Inspect the source code
- Deploy their own instance
- Modify the project

- Report bugs

Improve the implementation

For Shared Link Mode, public/shared links do not require MEGA account credentials.

For Account Mode, authentication information is involved. If using someone else's Worker, users should understand that they are trusting that deployment with the information sent to it.

For security-sensitive use, self-hosting and reviewing the source code provides greater transparency.

? THINGS TO KEEP IN MIND

The project depends on external services and infrastructure, so behaviour can change due to:

- MEGA changes
- Cloudflare limits
- API changes
- Network conditions
- Hosting policies
- Browser changes

Project updates

Important data should always have an independent backup.

? BASIC USAGE

- Create a Cloudflare Worker.
- Add the project Worker code.
- Deploy it.
- Open the Worker URL.
- Select the required mode.
- For Shared Link Mode, provide a public MEGA link.

Browse and access the required files.

The project repository contains the source code and deployment information.

? RELATED PROJECT: GOFILE INDEX

The developer also has another project called GoFile Index, which follows a related file-hosting/indexing concept.

Search for Quote:GoFile Index
if you are interested in similar projects.

? FINAL THOUGHTS

After testing MEGA INDEX CLOUDFLARE, I found the project interesting for users who frequently work with large MEGA shared folders or individual files.

The main concept is:

Code MEGA

?

CLOUDFLARE WORKER

?

FILE INDEX

?

HTTP-BASED ACCESS

Shared Link Mode is the simpler option because public/shared links do not require MEGA account credentials.

Account Mode provides additional functionality but requires more trust because authentication information is involved.

The experimental ZIP streaming feature is an interesting direction, but should currently be treated as work in progress.

Overall, I would describe the project as an open-source, self-hostable way to browse and access MEGA files through a Cloudflare Worker, with useful functionality already available and additional features under development.