

HTTP

The HTTP remote is a read only remote for reading files of a webserver. The webserver should provide file listings which rclone will read and turn into a remote. This has been tested with common web servers such as Apache/Nginx/Caddy and will likely work with file listings from most web servers. (If it doesn't then please file an issue, or send a pull request!)

Paths are specified as `remote:` or `remote:path/to/dir`.

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.
Choose a number from below, or type in your own value
[snip]
XX / http Connection
  \ "http"
[snip]
Storage> http
URL of http host to connect to
Choose a number from below, or type in your own value
  1 / Connect to example.com
    \ "https://example.com"
url> https://beta.rclone.org
Remote config
-----
[remote]
url = https://beta.rclone.org
-----
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
Current remotes:

Name                Type
====                ====
remote              http

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q

```

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

See all the top level directories

```
rclone lsd remote:
```

List the contents of a directory

```
rclone ls remote:directory
```

Sync the remote **directory** to **/home/local/directory**, deleting any excess files.

```
rclone sync -i remote:directory /home/local/directory
```

Read only

This remote is read only - you can't upload files to an HTTP server.

Modified time

Most HTTP servers store time accurate to 1 second.

Checksum

No checksums are stored.

Usage without a config file

Since the http remote only has one config parameter it is easy to use without a config file:

```
rclone lsd --http-url https://beta.rclone.org :http:
```

Standard Options

Here are the standard options specific to http (http Connection).

--http-url

URL of http host to connect to

- Config: url
- Env Var: RCLONE_HTTP_URL
- Type: string
- Default: ""
- Examples:
 - "https://example.com"
 - Connect to example.com
 - "https://user:pass@example.com"
 - Connect to example.com using a username and password

Advanced Options

Here are the advanced options specific to http (http Connection).

--http-headers

Set HTTP headers for all transactions

Use this to set additional HTTP headers for all transactions

The input format is comma separated list of key,value pairs. Standard [CSV encoding](#) may be used.

For example to set a Cookie use 'Cookie,name=value', or '"Cookie","name=value"'.

You can set multiple headers, e.g. '"Cookie","name=value","Authorization","xxx"'.

- Config: headers
- Env Var: RCLONE_HTTP_HEADERS
- Type: CommaSepList
- Default:

--http-no-slash

Set this if the site doesn't end directories with /

Use this if your target website does not use / on the end of directories.

A / on the end of a path is how rclone normally tells the difference between files and directories. If this flag is set, then rclone will treat all files with Content-Type: text/html as directories and read URLs from them rather than downloading them.

Note that this may cause rclone to confuse genuine HTML files with directories.

- Config: no_slash
- Env Var: RCLONE_HTTP_NO_SLASH
- Type: bool
- Default: false

--http-no-head

Don't use HEAD requests to find file sizes in dir listing

If your site is being very slow to load then you can try this option. Normally rclone does a HEAD request for each potential file in a directory listing to:

- find its size
- check it really exists
- check to see if it is a directory

If you set this option, rclone will not do the HEAD request. This will mean

- directory listings are much quicker
- rclone won't have the times or sizes of any files

- some files that don't exist may be in the listing
- Config: no_head
- Env Var: RCLONE_HTTP_NO_HEAD
- Type: bool
- Default: false

Limitations

`rclone about` is not supported by the HTTP 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.