

Koofr

Paths are specified as `remote:path`

Paths may be as deep as required, e.g. `remote:directory/subdirectory`.

The initial setup for Koofr involves creating an application password for rclone. You can do that by opening the Koofr [web application](#), giving the password a nice name like `rc1one` and clicking on generate.

Here is an example of how to make a remote called `koofr`. 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> koofr
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 / Koofr
    \ "koofr"
[snip]
Storage> koofr
** See help for koofr backend at: https://rclone.org/koofr/ **

Your Koofr user name
Enter a string value. Press Enter for the default ("").
user> USER@NAME
Your Koofr password for rclone (generate one at https://app.koofr.net/app/admin/preferences/p
y) Yes type in my own password
g) Generate random password
y/g> y
Enter the password:
password:
Confirm the password:
password:
Edit advanced config? (y/n)
y) Yes
n) No
y/n> n
Remote config
-----
[koofr]
type = koofr
baseurl = https://app.koofr.net
user = USER@NAME
password = *** ENCRYPTED ***
-----
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y

```

You can choose to edit advanced config in order to enter your own service URL if you use an on-premise or white label Koofr instance, or choose an alternative mount instead of your primary storage.

Once configured you can then use **rclone** like this,

List directories in top level of your Koofr

```
rclone ls koofr:
```

List all the files in your Koofr

```
rclone ls koofr:
```

To copy a local directory to an Koofr directory called backup

```
rclone copy /home/source remote:backup
```

Restricted filename characters

In addition to the [default restricted characters set](#) the following characters are also replaced:

Character	Value	Replacement
\	0x5C	\

Invalid UTF-8 bytes will also be [replaced](#), as they can't be used in XML strings.

Standard Options

Here are the standard options specific to koofr (Koofr).

--koofr-user

Your Koofr user name

- Config: user
- Env Var: RCLONE_KOOFR_USER
- Type: string
- Default: ""

--koofr-password

Your Koofr password for rclone (generate one at <https://app.koofr.net/app/admin/preferences/password>)

NB Input to this must be obscured - see [rclone obscure](#).

- Config: password
- Env Var: RCLONE_KOOFR_PASSWORD
- Type: string
- Default: ""

Advanced Options

Here are the advanced options specific to koofr (Koofr).

--koofr-endpoint

The Koofr API endpoint to use

- Config: endpoint
- Env Var: RCLONE_KOOFR_ENDPOINT
- Type: string
- Default: "https://app.koofr.net"

--koofr-mountid

Mount ID of the mount to use. If omitted, the primary mount is used.

- Config: mountid
- Env Var: RCLONE_KOOFR_MOUNTID
- Type: string
- Default: ""

--koofr-setmtime

Does the backend support setting modification time. Set this to false if you use a mount ID that points to a Dropbox or Amazon Drive backend.

- Config: setmtime
- Env Var: RCLONE_KOOFR_SETMTIME
- Type: bool
- Default: true

--koofr-encoding

This sets the encoding for the backend.

See: the [encoding section in the overview](#) for more info.

- Config: encoding
- Env Var: RCLONE_KOOFR_ENCODING
- Type: MultiEncoder
- Default: Slash,BackSlash,Del,Ctl,InvalidUtf8,Dot

Limitations

Note that Koofr is case insensitive so you can't have a file called "Hello.doc" and one called "hello.doc".