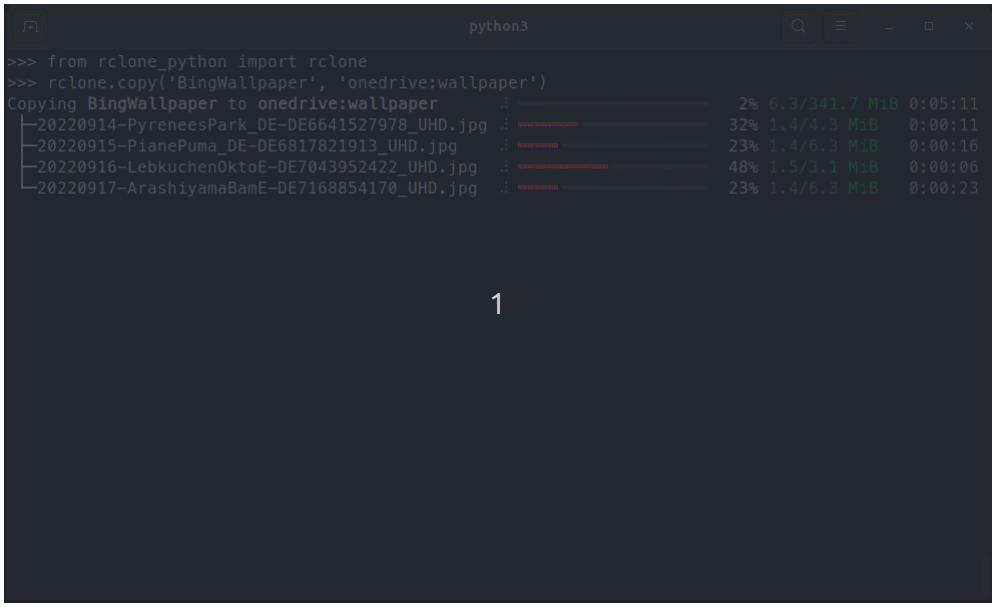
Description du projet

License MIT pypi package 0.1.19

rclone-python

A python wrapper for rclone that makes rclone's functionality usable in python.
rclone needs to be installed on the system for the wrapper to work.



Features

- Copy, move and sync files between remotes
- Delete and prune files/directories
- List files in a directory including properties of the files.
- List available remotes.
- Generate hashes from files or validate them with their hashsum.
- Create new remotes
- Check available remotes
- Create and manage public links.
- Check the currently installed rclone versions and if updates are available.

Installation

rclone_python can be installed using pip

```
pip install rclone-python
```

or by cloning this repository and running from within the root of the project

or by cloning this repository and running from within the root of the project

```
pip install .
```

How to use 💡

All functionally of this wrapper is accessible through `rclone`. The following example checks if rclone is installed.

```
from rclone_python import rclone

print(rclone.is_installed())
```

```
True
```

Create new remote

Create a new rclone remote connection with rclone's default client-id and client-secret.

```
from rclone_python import rclone
from rclone_python.remote_types import RemoteTypes

rclone.create_remote('onedrive', RemoteTypes.onedrive)
```

Additionally, client-id and client-secret can be used with many cloud providers.

```
from rclone_python import rclone
from rclone_python.remote_types import RemoteTypes

rclone.create_remote('onedrive', RemoteTypes.onedrive, client_id=
```

Copy

```
from rclone_python import rclone

# copy all file in the test_dir on OneDrive to the local data fol
rclone.copy('onedrive:data', 'data', ignore_existing=True, args=[
```

```
Copying onedrive:data to data ⋮ ──────────────────── ──────────────────
 ├─video1.webm                ⋮ ──────────── ──────────────────
 ├─video2.webm                ⋮ ──────────────────── ──────────────────
 └─video3.webm                ⋮ ──────────── ──────────────────
```

Delete

Delete a file or a directory. When deleting a directory, only the files in the directory (and all it's subdirectories) are deleted, but the folders remain.

```
from rclone_python import rclone

# delete a specific file on onedrive
rclone.delete('onedrive:data/video1.mp4')
```

Prune

```
from rclone_python import rclone

# remove the entire test_dir folder (and all files contained in i
rclone.purge('onedrive:test_dir')
```

Get Hash

```
from rclone_python import rclone
from rclone_python.hash_types import HashTypes

print(rclone.hash(HashTypes.sha1, "box:data")
```

```
{'video1.webm': '3ef08d895f25e8b7d84d3a1ac58f8f302e33058b', 'vide
```

Custom Progressbar

You can use your own rich progressbar with all transfer operations. This allows you to customize the columns to be displayed. A list of all rich-progress columns can be found here.

```
from rclone_python import rclone

from rich.progress import (
    Progress
```

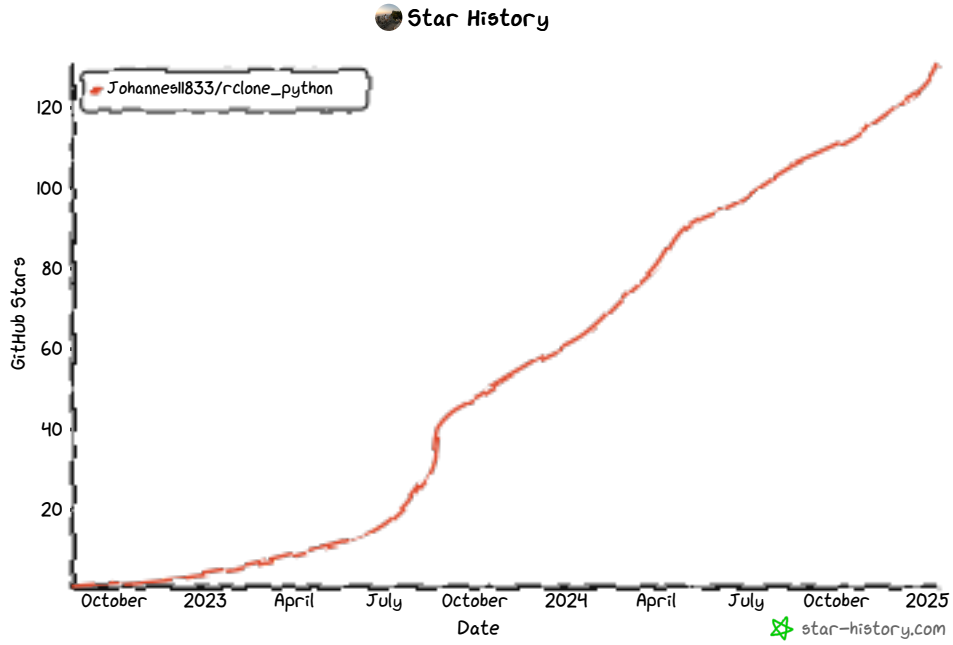
```
    Progress,
    TextColumn,
    BarColumn,
    TaskProgressColumn,
    TransferSpeedColumn,
)

pbar = Progress(
    TextColumn("[progress.description]{task.description}"),
    BarColumn(),
    TaskProgressColumn(),
    TransferSpeedColumn(),
)
rclone.copy("data", "box:rclone_test/data1", pbar=pbar)
```

```
Copying data to data1 ━━━━━━╺━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━  17% 5
 ├─video1.mp4         ━━━━━━━━━━━━━━╺━━━━━━━━━━━━━━━━━━━━━━  38% 4
 ├─video2.mp4         ━━╺━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━   5% 1
 └─another.mp4        ━━╺━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━   4% 1
```

Set the log level

The log level can be set using:

```
rclone.set_log_level(logging.DEBUG)
```

This will make the wrapper print the raw rclone progress.

Star History