

---

# **rmapy**

***Release 0.1***

**Stijn Van Campenhout**

**Dec 06, 2022**



# CONTENTS

<b>1</b>	<b>API Support</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Pip . . . . .	5
<b>3</b>	<b>Quick start</b>	<b>7</b>
3.1	quick start . . . . .	7
<b>4</b>	<b>API Documentation</b>	<b>11</b>
<b>5</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



This is an (unofficial) Remarkable Cloud API Client written in Python.



## API SUPPORT

List content in the cloud

Work with documents & folders

create a folder

move / rename a document or folder

create a document

edit a document

delete a document or folder

cli interface

export pdf with annotations





## INSTALLATION

### 2.1 Pip

Like any other package, you can install rmapy using pip:

```
pip install rmapy
```



## QUICK START

### 3.1 quick start

If you previously used the go package `rmapi`, the keys for authorization are re-used because we use the same storage location & format.

If not, you'll need to register the client as a new device on [my remarkable](#).

#### 3.1.1 Registering the API Client

Registering the device is easy. Go to [my remarkable](#) to register a new device and use the code you see on the webpage

```
1  from rmapi.api import Client
2
3  rmapy = Client()
4  # Should return False
5  rmapy.is_auth()
6  # This registers the client as a new device. The received device token is
7  # stored in the users directory in the file ~/.rmapi, the same as with the
8  # go rmapi client.
9  # Get a token at https://my.remarkable.com/device/desktop/connect.
10 rmapy.register_device("fkgzzklrs")
11 # It's always a good idea to renew the user token every time you start
12 # a new session.
13 rmapy.renew_token()
14 # Should return True
15 rmapy.is_auth()
```

#### 3.1.2 Working with items

The remarkable fs structure is flat containing metadata objects of two types:

- DocumentType
- CollectionType

We can list the items in the Cloud

```
1  >>> from rmapi.api import Client
2  >>> rmapy = Client()
3  >>> rmapy.renew_token()
4  True
```

(continues on next page)

(continued from previous page)

```

5 >>> collection = rmapy.get_meta_items()
6 >>> collection
7 <rmapy.collections.Collection object at 0x7fa1982d7e90>
8 >>> len(collection)
9 181
10 >>> # Count the amount of documents
11 ... from rmapy.document import Document
12 >>> len([f for f in collection if isinstance(f, Document)])
13 139
14 >>> # Count the amount of folders
15 ... from rmapy.folder import Folder
16 >>> len([f for f in collection if isinstance(f, Folder)])
17 42

```

## DocumentType

A DocumentType is a document. This can be a pdf, epub or notebook. These types are represented by the object `rmapy.document.Document`

Changing the metadata is easy

```

1 >>> from rmapy.api import Client
2 >>> rmapy = Client()
3 >>> rmapy.renew_token()
4 True
5 >>> collection = rmapy.get_meta_items()
6 >>> doc = [ d for d in collection if d.VissibleName == 'ModernC'][0]
7 >>> doc
8 <rmapy.document.Document a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3>
9 >>> doc.to_dict()
10 {'ID': 'a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3', 'Version': 1, 'Message': '', 'Success': True, 'BlobURLGet': '', 'BlobURLGetExpires': '0001-01-01T00:00:00Z', 'BlobURLPut': '', 'BlobURLPutExpires': '', 'ModifiedClient': '2019-09-18T20:12:07.206206Z', 'Type': 'DocumentType', 'VissibleName': 'ModernC', 'CurrentPage': 0, 'Bookmarked': False, 'Parent': ''}
11 >>> doc.VissibleName = "Modern C: The book of wisdom"
12 >>> # push the changes back to the Remarkable Cloud
13 ... rmapy.update_metadata(doc)
14 True
15 >>> collection = rmapy.get_meta_items()
16 >>> doc = [ d for d in docs if d.VissibleName == 'ModernC'][0]
17 Traceback (most recent call last):
18   File "<stdin>", line 1, in <module>
19 IndexError: list index out of range
20 >>> doc = [ d for d in docs if d.VissibleName == 'Modern C: The book of wisdom'][0]
21 >>> doc
22 <rmapy.document.Document a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3>
23 >>> doc.to_dict()
24 {'ID': 'a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3', 'Version': 1, 'Message': '', 'Success': True, 'BlobURLGet': '', 'BlobURLGetExpires': '0001-01-01T00:00:00Z', 'BlobURLPut': '', 'BlobURLPutExpires': '', 'ModifiedClient': '2019-09-18T20:12:07.206206Z', 'Type': 'DocumentType', 'VissibleName': 'Modern C: The book of wisdom', 'CurrentPage': 0, 'Bookmarked': False, 'Parent': ''}

```

## CollectionType

A CollectionType is a Folder.

These types are represented by the object `rmapy.folder.Folder`

Working with folders is easy!

```

1  >>> from rmapy.api import Client
2  >>> rmapy = Client()
3  >>> rmapy.renew_token()
4  True
5  >>> collection = rmapy.get_meta_items()
6  >>> collection
7  <rmapy.collections.Collection object at 0x7fc4718e1ed0>
8  >>> from rmapy.folder import Folder
9  >>> # Get all the folders. Note that the fs of Remarkable is flat in the cloud
10 ... folders = [ f for f in collection if isinstance(f, Folder) ]
11 >>> folders
12 [<rmapy.folder.Folder 028400f5-b258-4563-bf5d-9a47c314668c>, <rmapy.folder.Folder_
↳ 06a36729-f91e-47da-b334-dc088c1e73d2>, ...]
13 >>> # Get the root folders
14 ... root = [ f for f in folders if f.Parent == "" ]
15 >>> root
16 [<rmapy.folder.Folder 028400f5-b258-4563-bf5d-9a47c314668c>, <rmapy.folder.Folder_
↳ 5005a085-d7ee-4867-8859-4cd90dee0d62>, ...]
17 >>> # Create a new folder
18 ... new_folder = Folder("New Folder")
19 >>> new_folder
20 <rmapy.folder.Folder 579df08d-7ee4-4f30-9994-887e6341cae3>
21 >>> rmapy.create_folder(new_folder)
22 True
23 >>> # verify
24 ... [ f for f in rmapy.get_meta_items() if f.VissibleName == "New Folder" ]
25 [<rmapy.folder.Folder 579df08d-7ee4-4f30-9994-887e6341cae3>]
26 >>> [ f for f in rmapy.get_meta_items() if f.VissibleName == "New Folder" ][0].ID ==
↳ new_folder.ID
27 True
28 >>> # Move a document in a folder
29 ... doc = rmapy.get_doc("a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3")
30 >>> doc
31 <rmapy.document.Document a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3>
32 >>> doc.Parent = new_folder.ID
33 >>> # Submit the changes
34 ... rmapy.update_metadata(doc)
35 True
36 >>> doc = rmapy.get_doc("a969fcd6-64b0-4f71-b1ce-d9533ec4a2a3")
37 >>> doc.Parent == new_folder.ID
38 True

```

### 3.1.3 Uploading & downloading

reMarkable has a “special” file format for the raw documents. This is basically a zip file with files describing the document.

Here is the content of an archive retried on the tablet as example:

- 384327f5-133e-49c8-82ff-30aa19f3cfa40.content

- 384327f5-133e-49c8-82ff-30aa19f3cfa40-metadata.json
- 384326f5-133e-49c8-82ff-30aa19f3cfa40.pdf
- 384327f5-133e-49c8-82ff-30aa19f3cfa40.pagedata
- 384327f5-133e-49c8-82ff-30aa19f3cfa40.thumbnails/0.jpg
- 384327f5-133e-49c8-82ff-30aa19f3cfa40.highlights/9b75d8df-1d06-4c59-8f3e-4cf69aa96cd9.json

As the .zip file from remarkable is simply a normal .zip file containing specific file formats.

Highlights are stored in the `{uuid}.highlights/` folder.

You can find some help about the format at the following URL: <https://remarkablewiki.com/tech/filesystem>

## Uploading

To upload a pdf or epub file, we'll first need to convert it into the remarkable file format:

```
1 >>> from rmapy.document import ZipDocument
2 >>> from rmapy.api import Client
3 >>> rm = Client()
4 >>> rm.renew_token()
5 True
6 >>> rawDocument = ZipDocument(doc="/home/svancampenhout/27-11-2019.pdf")
7 >>> rawDocument
8 <rmapy.document.ZipDocument b926ffc2-3600-460e-abfa-0fcf20b0bf99>
9 >>> rawDocument.metadata["VissibleName"]
10 '27-11-2019'
```

Now we can upload this to a specific folder:

```
1 >>> books = [ i for i in rm.get_meta_items() if i.VissibleName == "Boeken" ][0]
2 >>> rm.upload(rawDocument, books)
3 True
```

And verify its existence:

```
1 >>> [ i.VissibleName for i in collection.children(books) if i.Type == "DocumentType" ]
↩
2 ['Origin - Dan Brown', 'Flatland', 'Game Of Thrones', '27-11-2019']
```

## API DOCUMENTATION

---

**Note:** This is an unofficial api client for the Remarkable Cloud. Use at your own risk.

---





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### r

`rmapy`, [11](#)



## INDEX

### R

rmapy (*module*), 11