

EXPORT DATA FROM RENTMAN

Because rentman does not allow to export your equipment data in a smart way, this script exports them smart as **json**, **md** and **PDF** to save locally or use it in your own frontend.

Specifically, the export formats provided by rentman frontend is .xls and .csv - those are missing one important aspect: The Files associated with an equipment (Images, PDFs, additional files).

The categories with which any equipment is associated is also added to the output.

The images are saved in best possible quality.

You'll need a JWT_TOKEN, accessible via rentman interface (Configuration > Extensions > API > Connect and then Show Token).

Put the token in a File named **JWT_TOKEN** (without file extension), put it in the same folder as the python script.

Start exporting with bash

This handy bash script guides you through the export process.

At first, it checks if you have an internet connection and also are entitled to access the rentman db.

You have the choice to export *everything*, *one or more specific items* by naming their **ID** from rentman; or define a *range*.

There is also the option to only download missing items.

```
./run_export.sh
```

When finished or aborted (**ctrl + c**) the scripts starts again so you can let this run until the sun consumes the earth.

Example output:

```
./run_export.sh  
Export EVERYTHING ? Y / N : n  
Export SPECIFIC articles? NUMBER,NUMBER,NUMBER,.. / N : 13,105  
? OVERWRITE existing files? Y / N : y  
Press CTRL and C to abort.  
  
-----  
Fetch 2 object(s) with code(s) [13, 105]  
  
Found 812 articles in DB.  
  
-----  
Collecting equipment data and creating files...  
103/812 ██████████ 12.7% - Gathering 6  
files and data of '33 10000666 Wardrobe - red'
```

```

    Fetched 2 object(s) with code(s) [13, 105] 🤔
    Process complete ✅

```

Test your connection to rentman manually

```
python3 checkAuth.py
```

will output **API call successful, yeah!** if it worked, or **API call failed** and some details about why.

Output


This script saves all your equipment and their files in a folder named "equipmentDump". In there, for each piece of equipment will be a folder created, named like this:

```
code_qrcodes_equipment name
```

eg.

```
33_10000666_Wardrobe - red
```

Its name is the "code" and the "qrcode" combined with the human readable title with umlaute and spaces. So don't name equipment with file system conflicting characters (i.e. don't use **&**, **/** or **'**)

 **Notice:** that the folder name does not include the **ID** of the item, as this is pretty much unusable for search in the rentman frontend and only used internally by them.

Additional files will be created within that folder. An example output with explanations:

```

equipmentDump/
├─ _archived/
├─ 33_10000666_Wardrobe - red/
│  └─ data.json
in a
│  │
way for further
│  │
shenanigans. This
│  │
additionally to the rentman
│  │
all the images/
│  │


```

- > contains all the data
- > computer-friendly
- > programming and
- > includes,
- > specs, data about
- > files associated

with the equipment as	
	> well as a human
readable category-path.	
─ data.md	> the same data in a
human readable and editable way.	
─ 33_10000666_Wardrobe - red.pdf	> PDF of data.md above:
Contains all the data in a	
	> humanreadable,
printable and non-editable way.	
	> Also includes a
link to the original file	
	> location stored on
an amazonS3 server by rentman.	
─ 33_10000666_Wardrobe - red-10000667-qr.svg	> QR codes images of
serial numbers if available	
─ 33_10000666_Wardrobe - red-10000668-qr.svg	> QR codes images of
serial numbers if available	
─ 33_10000666_Wardrobe - red-sheet.html	> A technical document
for generating the overview	
	> sheet PDFs
─ 33_10000666_Wardrobe - red-sheet.pdf	> Overview sheet as PDF
─ 78374rm4_photo-1.jpg	> attached image or
file	
─ 78374rm4_photo-2.jpg	> attached image or
file	
─ 78374rm4_photo-3.jpg	> attached image or
file	
─ 78374rm4_photo-4.jpg	> attached image or
file	
─ 78374rm4_photo-5.jpg	> attached image or
file	
─ 78374rm4_photo-6.jpg	> attached image or
file	
─ 34_10000789_Wardrobe - blue/	
─ 35_10000963_Hat stand ugly/	
─ ...	

Also, an "equipmentDump/_archived" is created which contains all archived equipment.

The folder "equipmentSheets/" is created in the root folder which holds copies of all the equipmentSheets for convenience.

 **Notice:** The speed is slowed to 10 calls a second to not trigger the safety feature by rentman API ("Not more than 20 calls / second"). So it takes a while to export everything.

Use the python script directly

For exporting everything:

```
python3 collectEverything.py
```

For exporting 10 articles, starting from article nr. 30:

```
python3 collectEverything.py --start 30 --num 10
```

For exporting one specific article, use arg `id`:

```
python3 collectEverything.py --id 200
```

This exports only the article with "Code" 200 ("Code" is what rentman calls their unique identifier).

Caveats

- Cannot be imported again to rentman
- "Tasks", "Notes" are not downloaded
- Label-PDFs associated with equipment is removed from the download. You can change that if you need!
- No special treatment for "sets" of other equipment
- "Custom fields" are not queryable, you'll need to change the list in the code to your liking if you want to output them properly.

Prerequisite for PDF conversion:

```
brew install basictex
```

```
brew install pandoc
```