

# Home Middleman API

Hubert Kasperek

March 4, 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	proxies . . . . .	2
1.2	file storage . . . . .	2
1.3	scraper . . . . .	2
1.4	clipboard . . . . .	3
1.5	notes . . . . .	3
1.6	tasks . . . . .	3
1.7	routine . . . . .	3
<b>2</b>	<b>TODO</b>	<b>4</b>
2.1	technical informations . . . . .	4
2.2	examples . . . . .	4

# 1 Introduction

## 1.1 proxies

Home Middleman has 4 types of proxy:

HTTP **/api/http**

HTTPS **/api/https**

TXT HTTP **/api/txt/http**

TXT HTTPS **/api/txt/https**

Http and https as the name suggests, just do http/https request and return the content of the html page, while the txt versions do the same, but instead of returning the html code, they return the text content of the website

*We can use proxies to use the internet more anonymously, more accessible to tools without page rendering options (txt proxy), or to collect html code or text from a website*

## 1.2 file storage

File listing **/api/files/list** allows you to list files in the *upload/* folder (this folder is used to store various types of files uploaded by the user or returned by the API)

File upload **/api/upload** allows you to upload files to the server

File upload from link **/api/uploadLink** allows you to upload files to the server by providing a link to a website containing a file, e.g. *https://example.com/image.jpg*

File write **/api/write** this api allows you to save content directly to a file of any format

File move **/api/files/mv** allows you to change the path to the file (only in the *upload/* folder area) or to change the name/format of the file

File delete **/api/files/del** allows you to delete files (only in the *upload/* folder area)

*File storage is basically a cloud, we can store files on it that we can access from any device in the network. Since the same folder as file storage is used by some scrapers, we can also have access to the scrapped data from any device*

## 1.3 scraper

Link scraper **/api/scrapper/links** allows you to save all links from the webpage in the *websitelink.json* format

Img scraper **/api/scrapper/imgs/** allows you to save all the available images from the webpage in the same format as they are on the web page

Cheerio html scraper **/api/scrapper/cheeriohtml** allows you to save any page elements in the .html format, the syntax of the Cheerio library is used to select page elements

*Scraper can be useful for collecting specific data of interest to us from html files hosted by other servers, in routine they can be extremely useful, for example, to collect news from the world every 24 hours*

## 1.4 clipboard

Get the latest saved clip **/api/clip** this api function is designed to return text, which is then to be used by home middleman clients to save this text to the clipboard (at least that's how it works in the Python client and the front-end of the application)

Save to clipboard **/api/clip/save** this api gives you the ability to save text to the clipboard for later use

Clipboard history **/api/clip/history** this api allows you to see history of saved clips in clipboard, to clear history you can use:

Clipboard erase **/api/clip/erase** this api clears all clipboard history including last saved text

*Clipboard fulfills its task fully only when hmm client has the option of saving the returned text to the clipboard*

## 1.5 notes

List notes **/api/notes** this api returns you saved notes

Add note **/api/notes/add** this api allows you to add a note

Remove note **/api/notes/del/** this api allows you to delete a note

*Notes require the following: title, note text, and date.*

## 1.6 tasks

Tasks listing **/api/task** this api returns the tasks saved on the server

Tasks returns listing **/api/task/log/** this api returns the values returned by tasks, e.g. if the scrapper is running and returns text instead of e.g. images or json, then this value is saved as a log entry

Count tasks **/api/task/count** this api returns the number of tasks on the server

Get the log entry at the specified index **/api/task/log/:num** this api returns one log entry, 0 is the first index, and 'n' is the newest one

Add new task **/api/task/add** this api allows you to add a new task

Delete task **/api/task/del/** this api allows you to delete a task

Start task **/api/task/run** allows you to start a task by specifying the name of the task you want to start

Start time task **/api/task/time/run** allows you to start a task by specifying the name of the task you want to start, but this one, unlike the previous one, is executed only after the time you specified

*Tasks are saved specific configurations of the use of other api's, therefore only they can be used in the routine.*

## 1.7 routine

Get information about the routine **/api/task/interval** returns information about tasks that are added to the routine (they are performed regularly according to the given time interval)

Add task to routine **/api/task/interval/add** allows you to add a task to the routine along with specifying how often (milliseconds by default, clients usually have it set to minutes) the given task should be performed

Delete task from routine **/api/task/interval/kill/** allows you to stop the routine of a given task

Count the tasks in the routine **/api/task/interval/count** this api function returns how many tasks run in intervals (routine)

*Routine allows you to perform tasks at intervals, so you don't have to manually perform a given activity from time to time - the software will do it for you!*

## 2 **TODO**

### 2.1 **technical informations**

### 2.2 **examples**