Winit

Webscrapper for Windows

User guide

Libor Polčák, Tomáš Kocman

TARZAN project VI20172020062 document
Faculty of Information Technology, Brno University of Technology

Last change: April 24, 2020

Winit — User guide

Libor Polčák, Tomáš Kocman

Faculty of Information Technology, Brno University of Technology, e-mail:
polcak@fit.vutbr.cz

This tool allows web page content scrapping and exporting the content as a compressed archive. The web crawl is performed using user-supplied regular expressions that may represent for example Torrent file names, Bitcoin wallets or keywords. Collected data may be used for law enforcement and other entitites, such as searching for information about a specific product or personal archive of web pages. The tool can be used in Microsoft Windows.

The aim of this document is to introduce the tool Winit[1] developed by *Integrated platform for analysis of digital data from security incidents* project.

1 Comparison of Winit and proof_platform

As the *Integrated platform for analysis of digital data from security incidents* project developed a similar tool — proof_platform, let us compare the two project.

Whereas proof_platform aims to be a fully scalable server-side option for web scrapping, Winit aims on Windows users.

Winit goals are following:

- Windows support,
- simple workflow to enable swift adoption by new users.

Unlike proof_platform, Winit is a monolithic application. There is just a single Scrapy process. To enable paralelism, the Scrapy process creates multiple Lemmiwinks[2] processes, each processes a URL and archives the web page. The processes are managed by apscheduler Python library.

2 Tool use case — What does a web page contains and what does it look like?

There are a lot of use cases during which an investigator is interested in a archive content of a web page. Winit aims to archive the exact look even if a part of the page is created by JavaScript.

Besides investigators, the tool can be handy to archive web pages for any purpose (provided that the archiver has legal grounds). Also, the tool is useful for researchers investigating trends in web development.

See the diploma thesis of Tomáš Kocman[3] for more use cases.

[1] https://gitlab.com/tomaskocman/winit
[2] https://www.fit.vut.cz/research/product/592/
[3] https://www.fit.vut.cz/study/thesis/21459/

3 Winit User Manual

3.1 Running the application

- Ensure you have activated the virtual environment with installed packages.
- Go to the `scrapitlite` directory.
- Run the application using scrapy crawl basic command in the terminal.

3.2 Description

Scrapy starts crawling process which crawls web pages, searches HTML for regex pattern and if a matching string is found, lemmitlite processes running in the background archives that web page. Archives are stored in `lemmitlite/output` directory.

3.3 Archives

Archives are in the MAFF format. The archive contains `index.htm` and `index_files` directory containing resources for used by the `index.htm` file. The MAFF archive can be treated as a zip file which means one can easily unzip the archive and open the archived web page.

3.4 Crawling settings

Configuration parameters are stored in `scrapitlite/basicplatform/settings.py`. This configuration file contains only the basic configuration parameters. For the whole list, go to the Scrapy settings manual page[4].

There are several mandatory parameters:

- `BOT_NAME` specifies the name of the crawling process.
- `SPIDER_MODULES, NEWSPIDER_MODULE` contains path to spiders.
- `ALLOWED_DOMAINS` specifies domains that crawler follows.
- `START_URLS` contains URLs for crawler to start.
- `PATTERN` specifies regular pattern for searching in HTML.
- `ITEM_PIPELINES` contains a list of pipeline modules (see online documentation[5]).

The apscheduler library is configured as follows:

- there are up to four Lemmiwinks processes,
- apscheduler stores all scrapping requests into a buffer and creates a new Lemmiwinks processes immediately after one of the previously scheduled processes finish.

The parameters can be changed in the `scrapitlite/basicplatform/pipelines/proxy.py` file.

[4] https://docs.scrapy.org/en/latest/topics/settings.html
[5] https://docs.scrapy.org/en/latest/topics/settings.html#item-pipelines