coskundeniz / ad_clicker

Code Issues Pull requests Actions Projects Wiki Security Insights

Google Ads clicker

coskundeniz.github.io/ad_clicker/

GPL-3.0 license

104 stars 49 forks 8 watching 2 Branches 22 Tags Activity

Public repository

main 2 Branches 22 Tags Go to file t Go to file Add file Code

coskundeniz Update undetected-chromedriver version last week

.github	Add new links	2 years ago
assets	Update help image for wiki page	4 months ago
docs	Separate website for premium	4 months ago
.dockerignore	Add GitHub page for the project	last year
.gitignore	Fix race condition in driver patching for ...	6 months ago
Dockerfile	Add Docker image	last year
LICENSE	Initial commit	3 years ago
README.md	Add max_scroll_limit option from premiu...	2 weeks ago
ad_clicker.py	Add max_scroll_limit option from premiu...	2 weeks ago
config.py	Add poems to modules(just for fun)	8 months ago
geolocation_db.py	Add poems to modules(just for fun)	8 months ago
proxy.py	Fix import error seen on some Windows ...	2 months ago
requirements.txt	Update undetected-chromedriver version	last week
run_ad_clicker.py	Add max_scroll_limit option from premiu...	2 weeks ago
run_in_loop.py	Add max_scroll_limit option from premiu...	2 weeks ago
sample_queries.txt	Add an example queries file	9 months ago
search_controller.py	Add max_scroll_limit option from premiu...	2 weeks ago
utils.py	Add new Chrome options, user agents, a...	2 months ago

README GPL-3.0 license

ad_clicker

This command-line tool clicks ads for a certain query on Google search using undetected_chromedriver package. Supports proxy, running multiple simultaneous browsers, ad targeting/exclusion, and running in loop.

Old version of the tool can be found in the `old_version` branch.

- Requires Python 3.9+
- Requires Chrome latest version

[Related post on Medium]

How to setup

Run the following commands in the project directory to install the required packages.

- `python -m venv env`
- `source env/bin/activate`
- `python -m pip install -r requirements.txt`

See [here] for setup on Windows.

See [here] for creating the Docker image manually.

How to run

- You need to see `(env)` at the beginning of your command prompt that is showing virtual environment is activated.
- Before running the below commands for the first time, run `python ad_clicker.py -q test` once and end it by pressing CTRL+C after seeing the browser opened.

```
usage: python ad_clicker.py [-h] [-q QUERY] [-e EXCLUDES] [-l MAX_SCROLL_LIMIT] [--headless] [-p PROXY]
[-pf PROXY_FILE] [--auth] [--incognito]

optional arguments:
  -h, --help                                                show this help message and exit
  -q QUERY, --query QUERY                                   Search query
  -l MAX_SCROLL_LIMIT, --max_scroll_limit MAX_SCROLL_LIMIT  Number of maximum scrolls on the
search results page
  --headless                                                Use headless browser (not
recommended)
  -p PROXY, --proxy PROXY                                   Use the given proxy in ip:port format
  -pf PROXY_FILE, --proxy_file PROXY_FILE                   Select a proxy from the given file
  --auth                                                    Use proxy with username and password.
If this is passed,
                                                            proxy parameter should be in
"username:password@host:port" format
  -e EXCLUDES, --excludes EXCLUDES                          Exclude the ads that contain given
words in url or title
  --incognito                                               Run in incognito mode. Disables proxy
usage.
  -qf QUERY_FILE, --query_file QUERY_FILE                   Read queries to search from the given
file (valid for multiprocess run)
  -bc BROWSER_COUNT, --browser_count BROWSER_COUNT          Maximum number of browsers to run
```

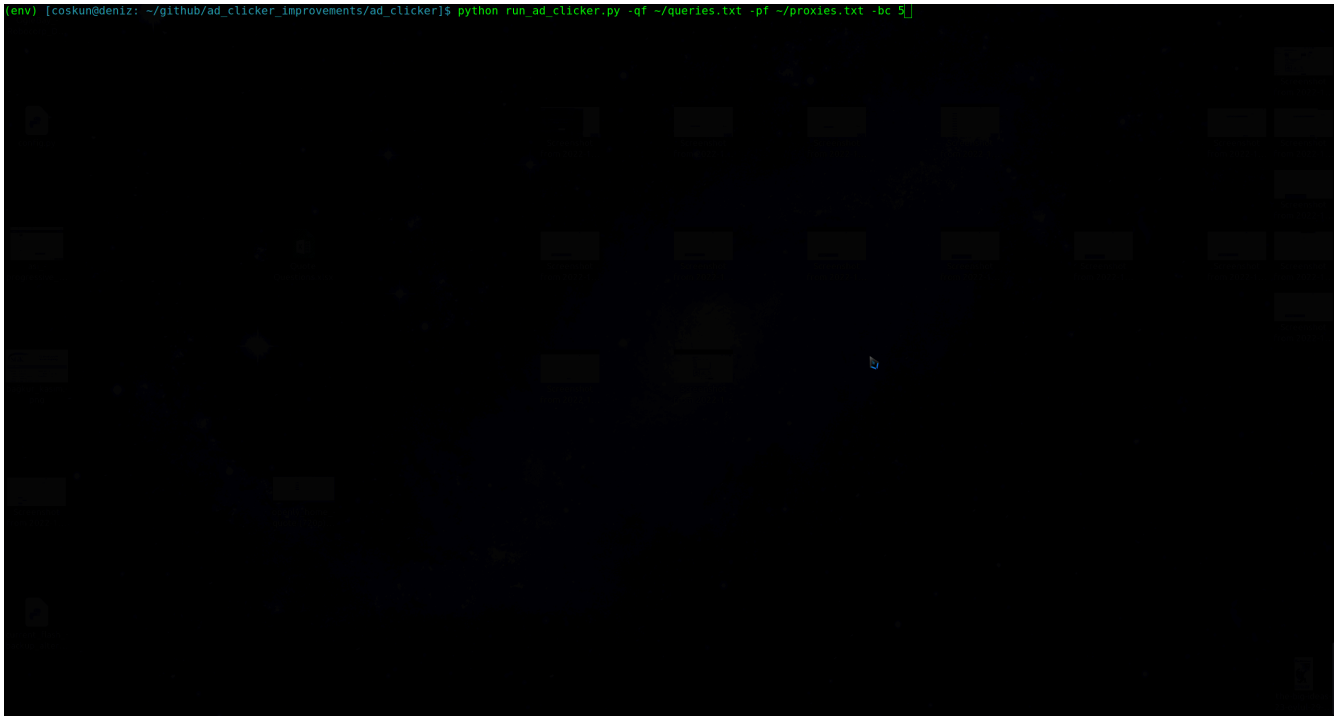
```
  concurrently (valid for multiprocess run)
    -ms MULTIPROCESS_STYLE, --multiprocess_style MULTIPROCESS_STYLE  Style of the multiprocess run. (valid
  for multiprocess run)
                                                                     1: different query on each browser
  (default)
                                                                     2: same query on each browser


 python ad_clicker.py -q <search query> [-e EXCLUDES] [-l MAX_SCROLL_LIMIT] [--headless] [-p PROXY] [-pf
PROXY_FILE] [--auth] [--incognito]
```

Examples

```
"~/" represents home folder on Linux/Mac. If you use Windows, don't use it.
If you put your files into project directory, you can just specify the file name,
otherwise you should give the full path.
```

- Search for "wireless keyboard" with the default 4 seconds visit time on clicked ad pages.
  - `python ad_clicker.py -q "wireless keyboard"`
- Search for "wireless keyboard" with maximum scroll set to 5.
  - `python ad_clicker.py -q "wireless keyboard" -l 5`
  - By default(0), it will scroll until the end.
- Search for "wireless keyboard" using headless browser.
  - `python ad_clicker.py -q "wireless keyboard" --headless`
- Search for "wireless keyboard" using the given proxy without authentication.
  - `python ad_clicker.py -q "wireless keyboard" -p host:port`
- Search for "wireless keyboard" using the given proxy with authentication.
  - `python ad_clicker.py -q "wireless keyboard" --auth -p username:password@host:port`
- Search for "wireless keyboard" using a proxy from the given file.
  - `python ad_clicker.py -q "wireless keyboard" -pf ~/proxies.txt`
- Search for "wireless speaker" and click links that include the given filter words in url or title.
  - `python ad_clicker.py -q "wireless speaker@amazon#ebay  # mediamarkt"`
  - Spaces around "@" and "#" are ignored, so both "wireless speaker@amazon#ebay" and "wireless speaker @ amazon # ebay" take "wireless speaker" as search query and "amazon" and "ebay" as filter words.
  - If you will give a target domain as filter word, don't use "http" or "www" parts in it. Use like "query@domainname.com" or even "query@domainname". Keep it as short as possible to get a match.
- Search for "wireless speaker" and click links except the ones containing the given words in url or title.
  - `python ad_clicker.py -q "wireless speaker" -e "amazon.com,mediamarkt.com,for 2022,Soundbar"`
  - Separate multiple exclude items with comma.
- Run browser in `incognito` mode.

    - `python ad_clicker.py -q "wireless speaker" --incognito`
    - Note that the proxy extension is not enabled in *incognito* mode.
- Run multiple browsers by taking queries and proxies with authentication from the given files.
    - `python run_ad_clicker.py -qf ~/queries.txt --auth -pf ~/proxies.txt`
    - You can see example queries in `sample_queries.txt` file.
- Run multiple browsers by taking queries and proxies with authentication from the given files and pass exclude words.
    - `python run_ad_clicker.py -qf ~/queries.txt --auth -pf ~/proxies.txt -e "amazon.com,mediamarkt.com,for 2022"`
- Run multiple browsers by taking queries and proxies without authentication from the given files with 5 browsers.
    - `python run_ad_clicker.py -qf ~/queries.txt -pf ~/proxies.txt -bc 5`
    - If -bc(--browser_count) option is not given, the number of cpu cores is used.
- Run multiple browsers by taking queries and proxies from the given files using alternative multiprocess style.
    - `python run_ad_clicker.py -qf ~/queries.txt -pf ~/proxies.txt -ms 2`
    - **1**: each browser instance gets a different query from file (default) (e.g. 5 browsers search the first 5 queries from the file. After they are completed, second group of 5 browsers search the next 5 queries from the file and so on)
    - **2**: multiple browser instances get the same query (e.g. 5 browsers search the first query from file. After they are completed, second group of 5 browsers search the second query and so on)
    - If the number of queries or proxies are less than the number of browsers to run, they are cycled.



- Run the tool in loop.
    - `python run_in_loop.py -qf ~/queries.txt -pf ~/proxies.txt --auth`

  - Query and proxy files are required for this command.
  - Default wait time between runs is 60 seconds.
- Run the tool in loop with given number of browser instances.
  - `python run_in_loop.py -qf ~/queries.txt -pf ~/proxies.txt --auth -bc 4`
- Run the tool in loop with given number of browser instances and maximum scroll set to 10.
  - `python run_in_loop.py -qf ~/queries.txt -pf ~/proxies.txt --auth -bc 4 -l 10`
- Run the tool in loop with given number of browser instances and 2 minutes wait between runs.
  - `python run_in_loop.py -qf ~/queries.txt -pf ~/proxies.txt --auth -bc 2 -wt 120`
  - Wait time parameter should be in seconds.
- Run the tool in loop with given number of browser instances and pass exclude words.
  - `python run_in_loop.py -qf ~/queries.txt -pf ~/proxies.txt --auth -bc 2 -e "amazon.com,mediamarkt.com,for 2022"`
- Run the tool in loop using alternative multiprocess style.
  - `python run_in_loop.py -qf ~/queries.txt -pf ~/proxies.txt --auth -ms 2`

How to run Docker image on Linux

- If you want to use the prebuilt image from DockerHub, you can use the following command.
  - `docker pull codenineeight/ad_clicker_docker`
- Run the following command on your host machine once.
  - `xhost local:docker`
- Create a volume for the input files
  - `docker volume create ad_clicker_files`
- Add your queries and proxies files to the created volume
  - `docker run -v ad_clicker_files:/data -v <queries.txt folder path>:/mnt/host alpine cp /mnt/host/queries.txt /data/`
  - `docker run -v ad_clicker_files:/data -v <proxies.txt folder path>:/mnt/host alpine cp /mnt/host/proxies.txt /data/`
  - Check if the files are copied successfully

```
docker run --rm -v ad_clicker_files:/data alpine ls -l /data
```

```
-rw-r--r--    1 root     root           165 Apr  1 09:25 proxies.txt
-rw-r--r--    1 root     root           216 Apr  1 09:25 queries.txt
```

- You can pass the arguments as before after the image name(in this case codenineeight/ad_clicker_docker)

- `docker run --rm -it --net=host -e DISPLAY=$DISPLAY -v ad_clicker_files:/data codenineeight/ad_clicker_docker run_ad_clicker.py -qf /data/queries.txt --auth -pf /data/proxies.txt -bc 2`
- `docker run --rm -it --net=host -e DISPLAY=$DISPLAY codenineeight/ad_clicker_docker ad_clicker.py -q "wireless keyboard" -p username:password@host:port --auth`

Support & Premium Version

https://coskundeniz.github.io/ad_clicker

If you benefit from this tool, please consider donating using the sponsor links.

Releases 22

v1.1.21 Latest
last week

+ 21 releases

Sponsor this project

patreon.com/**pythondoctor**

ko-fi.com/**coskundeniz**

liberapay.com/**coskundeniz**

https://www.buymeacoffee.com/coskundeniz

Packages

No packages published

Deployments 38

github-pages last week

+ 37 deployments

Languages

Python 98.5% **Dockerfile** 1.5%