

```

1  # -*- coding: cp1251 -*-
2  import os
3  import threading
4  import time
5  from multiprocessing import Pool
6
7  import undetected_chromedriver as uc
8  from selenium.webdriver.remote.webdriver import By
9
10 my_path = f"{os.getcwd()}\\CIAN_diplom"
11 whole_time = time.time()
12
13
14 def create_urls() -> list:
15     """
16     Create list of URLs for all pages with different JKs to use multiprocessing easily
17     """
18     global url_list
19     url_list = []
20
21     with open(rf"{my_path}\\page_number_dict2.txt", "r", encoding="utf-8") as f:
22
23         for line in f:
24             url, pages = line.rstrip().split(": ")
25             url = url[:-1]
26             pages = int(pages)
27
28             for page in range(1, pages + 1):
29                 full_url = f"{url}{page}"
30                 url_list.append(full_url)
31
32     return url_list
33
34
35 threadLocal = threading.local()
36
37
38 def get_driver() -> None:
39     """
40     Create webdriver and reassign attributes to use same driver multiple times
41     """
42     driver = getattr(threadLocal, "driver", None)
43     chromeOptions = uc.ChromeOptions()
44     chromeOptions.add_argument("--headless")
45     if driver is None:
46         driver = uc.Chrome(chrome_options=chromeOptions)
47     setattr(threadLocal, "driver", driver)
48     return driver
49
50
51 def get_urls(url: str) -> None:
52     driver = get_driver()
53     try:
54         url_time = time.time()
55         full_url = url
56         driver.get(full_url)
57         driver._web_element_cls = uc.UCWebElement
58
59         href_raw = driver.find_elements(
60             By.XPATH,
61             f'//*[@id="frontend-serp"]/div/div[@class="_93444fe79c--wrapper--W0WqH"
62             f'and not(@data-name="Suggestions")]/article[@data-name="CardComponent"]/div[1]/div[2]/div[1]/div/a',
63         )
64
65         ad = len(href_raw)
66
67         with open(rf"{my_path}\\flat_final_urls.txt", "a") as ouf:
68             for item in href_raw:
69                 ouf.write(str(item.get_attribute("href")))
70                 ouf.write("\n")
71
72         with open(rf"{my_path}\\flat_final_urls.txt", "r") as f:
73             count_ad = sum(1 for _ in f)
74
75         print(
76             f"Adverts {ad}, {full_url[-12:]}, {(time.time()-url_time)//60:.0f}min {(time.time()-url_time)%60:.2f}s; "
77             f"in total {count_ad}, {(time.time()-whole_time)//60:.0f}min {(time.time()-whole_time)%60:.2f}s"
78         )
79
80     except Exception as ex:
81         print(ex)
82
83
84 if __name__ == "__main__":
85     create_urls()

```

```
86     p = Pool(processes=3)
87     p.map(get_urls, url_list)
```