A to Z Databases Web Scraper

August 10, 2024

[1]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException
from selenium.common.exceptions import ElementClickInterceptedException
```

[4]:
```
from selenium import webdriver

driver = webdriver.Chrome()
#driver.get('https://www.google.com')
```

[ ]:
```
from selenium import webdriver
import selenium
import requests
from bs4 import BeautifulSoup

from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
from selenium.common.exceptions import NoSuchElementException

chromedriver_path = 'C:/Users/Administrator/Downloads/chromedriver_win32/
 ↪chromedriver.exe'
chrome_options = webdriver.ChromeOptions()
driver_service = Service(chromedriver_path)
#driver = webdriver.Chrome(service=driver_service, options=chrome_options)
driver = webdriver.Chrome()
AtoZ_url = 'https://ezproxy.nmls.lib.tx.us/login?url=http://www.atozdatabases.
 ↪com'
driver.get(AtoZ_url)

barcode_input = driver.find_element(By.XPATH, '/html/body/form/p[1]/input')
pin_input = driver.find_element(By.XPATH, '/html/body/form/p[2]/input')
```

```
barcode_value = '243127'
pin_value = '5123'

barcode_input.send_keys(barcode_value)
pin_input.send_keys(pin_value)

submit_button = driver.find_element(By.XPATH, '/html/body/form/p[3]/font/input')
submit_button.click()

wait = WebDriverWait(driver, 10)
get_started_button = driver.find_element(By.XPATH, '//*[@id="total-wrapper"]/
 ↪div/div/div[3]/div[2]/div/div/div[2]/div[2]/div/div[2]/div[3]/a/span')
get_started_button.click()
```

[7]:
```
state_button = driver.find_element(By.XPATH, '//*[@id="jQMenuPortletId"]/div[2]/
 ↪div[1]/div[2]/div[1]/label')
state_button.click()

# wait = WebDriverWait(driver, 10)  # wait for up to 10 seconds
# newjersey_button = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="ld31"]')))
# newjersey_button.click()

wait = WebDriverWait(driver, 10)  # wait for up to 10 seconds
ohio_button = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="ld36"]')))
ohio_button.click()

# wait = WebDriverWait(driver, 10)  # wait for up to 10 seconds
# pennsylvania_button = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="ld39"]')))
# pennsylvania_button.click()

headofhousehold_button = driver.find_element(By.XPATH, '//
 ↪*[@id="jQMenuPortletId"]/div[2]/div[5]/div[2]/div[3]/label')
headofhousehold_button.click()

wait = WebDriverWait(driver, 10)  # wait for up to 10 seconds
headonly_element = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="label_Ind_Household_Rank_Code0"]')))
headonly_element.click()

homeownerORrenter_button = driver.find_element(By.XPATH, '//
 ↪*[@id="jQMenuPortletId"]/div[2]/div[6]/div[2]/div[6]/label')
homeownerORrenter_button.click()
```

```
wait = WebDriverWait(driver, 10)  # wait for up to 10 seconds
homeowner_element = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="label_Home_Owner_Renter_Code0"]')))
homeowner_element.click()

file_path = 'C:/Users/Administrator/Desktop/Area Codes/Selected Area Codes OH.
 ↪txt'

with open(file_path, 'r') as file:
    area_codes = file.read()

areacode_button = driver.find_element(By.XPATH, '//*[@id="jQMenuPortletId"]/
 ↪div[2]/div[4]/div[2]/div[2]/label')
areacode_button.click()

wait = WebDriverWait(driver, 10)  # wait for up to 10 seconds
pastecodes_element = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="toggleicon_Area_Code"]')))
pastecodes_element.click()

input_box = driver.find_element(By.XPATH, '//*[@id="textareaPaste_Area_Code"]')
input_box.send_keys(area_codes)

wait = WebDriverWait(driver, 20)  # wait for up to 40 seconds
search_button = wait.until(EC.element_to_be_clickable((By.XPATH, '//
 ↪*[@id="total-count-wrapper"]/div/div[1]/a/span')))
search_button.click()
```

[5]:
```
dropdown_button = driver.find_element(By.XPATH, '//*[@id="recordFilter"]/
 ↪option[3]')
dropdown_button.click()
```

[10]:
```
import time
page_input_xpath = '//*[@id="resultFormId"]/div[1]/div[1]/div[3]/div/div[1]/
 ↪div[2]/input'
select_all_xpath = '//*[@id="checkall"]'
forward_page_button_xpath = '//*[@id="resultFormId"]/div[1]/div[1]/div[3]/div/
 ↪div[1]/div[3]'
overlay_xpath = '//*[@id="loading_image"]'
download_button_xpath = '//*[@id="resultFormId"]/div[1]/div[1]/div[4]/div/a[6]/
 ↪span'
file_name_input_xpath = '//*[@id="_customName"]'
continue_button_xpath = '/html/body/div[5]/div[11]/div/button[2]/span'
revise_search_button_xpath = '//*[@id="resultFormId"]/div[1]/div[1]/div[2]/
 ↪div[2]/a'
```

```
search_button_xpath = '//*[@id="total-count-wrapper"]/div/div[1]/a/span'

current_page = 2001

def select_1000_rows():
    global current_page
    for _ in range(10):
        # Input the desired page number
        page_input_box = WebDriverWait(driver, 30).until(EC.
 ↪element_to_be_clickable((By.XPATH, page_input_xpath)))
        page_input_box.clear()
        page_input_box.send_keys(str(current_page))
        page_input_box.send_keys(Keys.RETURN)

        try:
            WebDriverWait(driver, 20).until(EC.
 ↪invisibility_of_element_located((By.XPATH, overlay_xpath)))
        except TimeoutException:
            pass

        select_all_box = WebDriverWait(driver, 30).until(EC.
 ↪element_to_be_clickable((By.XPATH, select_all_xpath)))
        select_all_box.click()

        if _ != 9:  # No need to click forward on the last iteration
            forward_page_button = WebDriverWait(driver, 20).until(EC.
 ↪element_to_be_clickable((By.XPATH, forward_page_button_xpath)))
            forward_page_button.click()

        current_page += 1  # Move to the next page

def download_selected_rows():

    try:
        WebDriverWait(driver, 40).until(EC.invisibility_of_element_located((By.
 ↪XPATH, overlay_xpath)))
    except TimeoutException:
        print("Warning: Overlay might still be present!")

    download_button = WebDriverWait(driver, 40).until(EC.
 ↪element_to_be_clickable((By.XPATH, download_button_xpath)))
    download_button.click()

    try:
        WebDriverWait(driver, 40).until(EC.invisibility_of_element_located((By.
 ↪XPATH, overlay_xpath)))
```

```
    except TimeoutException:
        print("Warning: Overlay might still be present after clicking download!
↪")

    file_name_input = WebDriverWait(driver, 40).until(EC.
↪element_to_be_clickable((By.XPATH, file_name_input_xpath)))
    current_file_name = f'OH {int(current_page / 10)}'
    file_name_input.clear()  # Clear any existing value
    file_name_input.send_keys(current_file_name)

    # Check and confirm the file name
    print(f"Setting file name to: {current_file_name}")

    continue_button = WebDriverWait(driver, 40).until(EC.
↪presence_of_element_located((By.XPATH, continue_button_xpath)))
    driver.execute_script("arguments[0].scrollIntoView(true);", continue_button)

    continue_button = WebDriverWait(driver, 40).until(EC.
↪element_to_be_clickable((By.XPATH, continue_button_xpath)))
    continue_button.click()

    time.sleep(5)

    #continue_button_timeout = 60

    #try:
       # continue_button = WebDriverWait(driver, continue_button_timeout).
↪until(EC.presence_of_element_located((By.XPATH, continue_button_xpath)))
        #driver.execute_script("arguments[0].scrollIntoView(true);",␣
↪continue_button)

        #continue_button = WebDriverWait(driver, continue_button_timeout).
↪until(EC.element_to_be_clickable((By.XPATH, continue_button_xpath)))
        #continue_button.click()

    #except TimeoutException:
        #raise TimeoutException(f"Could not find the 'Continue' button using␣
↪XPATH '{continue_button_xpath}' within {continue_button_timeout} seconds.")

    #time.sleep(5)

def click_element_with_retry(xpath, max_retries=3):
    for _ in range(max_retries):
        try:
            element = WebDriverWait(driver, 40).until(EC.
↪presence_of_element_located((By.XPATH, xpath)))
```

```
            driver.execute_script("arguments[0].click();", element)
            return  # Exit if the JavaScript click was successful
        except (ElementClickInterceptedException,␣
 ↪StaleElementReferenceException):
            time.sleep(1)  # Wait for a second before retrying
    raise Exception(f"Failed to click on element after {max_retries} attempts.")

def revise_search_and_initiate_new_search():

    click_element_with_retry(revise_search_button_xpath)
    click_element_with_retry(search_button_xpath)

# Main script execution for 10 sets of 1000 rows
for _ in range(10):
    select_1000_rows()
    download_selected_rows()
    if _ != 9:  # No need to revise search after the last batch
        revise_search_and_initiate_new_search()
```

```
Setting file name to: OH 201
Setting file name to: OH 202
Setting file name to: OH 203
Setting file name to: OH 204
Setting file name to: OH 205
Setting file name to: OH 206
Setting file name to: OH 207
Setting file name to: OH 208
Setting file name to: OH 209
Setting file name to: OH 210
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: