import time

from datetime import datetime

from selenium import webdriver

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.by import By

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.ui import WebDriverWait

from webdriver\_manager.chrome import ChromeDriverManager

"""

########### IMPLICIT WAIT ###########

Aplicatiile web pot avea mai multe elemente dinamice care apar/dispar in functie de anumite actiuni ale utilizatorului,

sau pot fi cazuri in care timpul de incarcare/randarea site-ului sunt ineficiente, iar elementele nu apar instant.

Daca driverul incearca sa interactioneze cu astfel de elemente inainte ca acestea sa apara, se va returna o eroare,

iar testele vor fi marcate ca fiind "picate".

Funcția implicitly\_wait() din Selenium WebDriver specifică un timp maxim de așteptare implicită pentru căutarea unui

element pe pagină înainte ca WebDriver-ul să arunce o excepție NoSuchElementException.

Pe scurt, WebDriver-ul va aștepta pentru o perioadă specificată înainte de a arunca o excepție în cazul în care

elementul nu este găsit instant.

ATENTIE: Odata setat, implicitly\_wait() se va aplica pe tot parcursul "vietii" driverului sau pana cand acesta va fi

schimbat cu un alt implicitly\_wait() aplicat pe driver.

"""

# declaram un serviciu - va instala browserul in cazul in care nu exista pe PC

service = Service(ChromeDriverManager().install())

driver = webdriver.Chrome(service=service)

LINK = "https://chercher.tech/practice/explicit-wait-sample-selenium-webdriver"

driver.get(LINK)

# Functie ajutatoare - ne returneaza ora curenta in formatul ora:minut:secunda

def get\_time():

return datetime.now().strftime('%H:%M:%S')

# ### EXERCITIU 1:

# - Click pe butonul "Change Text to Selenium Webdriver"

# - Dupa 10 secunde textul "site" se va schimba in "Selenium Webdriver"

# - Vom folosi un implicit wait pentru a face driverul sa astepte maxim 11 secunde inainte sa dea eroare

# ## exemplu negativ:

# cu un wait de 5 secunde testul va pica, deoarece elementul devine prezent dupa 10 secunde

# driver.implicitly\_wait(5)

# button\_change\_text = driver.find\_element(By.ID, "populate-text")

# button\_change\_text.click()

# h2\_text\_selenium\_webdriver = driver.find\_element(By.XPATH, "//h2[@id='h2' and text()='Selenium Webdriver']")

driver.implicitly\_wait(11)

button\_change\_text = driver.find\_element(By.ID, "populate-text")

button\_change\_text.click()

print(f"{get\_time()}: Incepem asteptarea")

h2\_text\_selenium\_webdriver = driver.find\_element(By.XPATH, "//h2[@id='h2' and text()='Selenium Webdriver']")

print(f"{get\_time()}: S-a gasit elementul")

assert h2\_text\_selenium\_webdriver.is\_displayed()

# Resetam wait-ul implicit la 0

driver.implicitly\_wait(0)

"""

########### EXPLICIT WAITS ###########

Wait-urile explicite sunt putin diferite de implicit\_wait(), deoarece ele se aplica o singura data (in momentul in care

sunt apelate ca functie) existand mai multe conditii dupa care putem astepta sa fie indeplinite.

Exemple de conditii:

- prezenta elementului

- vizibilitatea elementului

- elementul sa contina un text

- atributul unui element sa existe

- atributul unui element sa contina o anumita valoare

- etc...

Structura unui explicit wait:

wait = WebdriverWait(driver, x)

wait.until(expected\_condition)

unde:

\* WebdriverWait - este clasa care face driverul sa astepte

\* x - este cantitatea maxima de timp pana la indeplinirea conditiei (si inaintea returnarii unei erori)

\* expected\_condition - conditia dupa care asteptam sa se indeplineasca

"""

# Declaram WebElement pentru butonul care va afisa un alt buton dupa 10 secunde

button\_display\_button = driver.find\_element(By.ID, "display-other-button")

# Click pe buton

button\_display\_button.click()

# Declaram un WebdriverWait de 10 secunde - aici doar definim wait-ul, asteptarea inca nu incepe

wait = WebDriverWait(driver, 11)

# Deoarece butonul este prezent in DOM, dar este invizibil, va trebui sa asteptam pana cand acesta devine vizibil

print(f"{get\_time()}: Incepem asteptarea")

# aici incepem sa asteptam dupa butonul care trebuie sa devina vizibil

wait.until(EC.visibility\_of\_element\_located((By.ID, "hidden")))

print(f"{get\_time()}: Elementul a devenit vizibil")

driver.find\_element(By.ID, "hidden").click()