**Deneme 9  
Tarih: 14/11/2024**

**Ad: MOHAMMED MASHHOR ALMASHHOR**

**NO: 9211118091**

**Dosya Yükleme ve İndirme Testleri**

**Dosya yükleme işlemleri:**

* send\_keys() ile dosya yükleme
* HTML input[type='file'] etiketleriyle çalışma

**Dosya indirme işlemleri:**

* Tarayıcı indirme ayarlarının yapılandırılması
* İndirilen dosyanın varlığını ve içeriğini doğrulama

**1.Dosya Yükleme İşlemleri**

**send\_keys() ile Dosya Yükleme**

send\_keys() yöntemi, dosya yükleme için kullanılan en yaygın yöntemdir. input[type='file'] etiketi varsa, send\_keys() ile dosya yolu göndererek dosyayı yükleyebiliriz.

from selenium import webdriver

from selenium.webdriver.common.by import By

# WebDriver başlatma

driver = webdriver.Chrome()

# Test edilecek sayfaya git

driver.get("URL")

# Dosya yükleme elementini bul ve send\_keys() ile dosya yolunu gönder

file\_input = driver.find\_element(By.XPATH, "//input[@type='file']")

file\_input.send\_keys("/path/to/your/file.txt")

# Doğrulama işlemleri (sayfanın dosya yüklendiğini göstermesi veya dosyanın listeye eklenmesi gibi)

**b. HTML input[type='file'] Etiketleriyle Çalışma**

Yukarıdaki örnek aslında input[type='file'] etiketleriyle çalışmaya örnektir. Eğer bir dosya yükleme kutusunu bulup send\_keys() kullanamıyorsanız, başka bir yöntem kullanmak gerekebilir. Ancak, input[type='file'] elementi görünmezse JavaScript ile görünür yapılması gerekebilir.

**2. Dosya İndirme İşlemleri**

**a. Tarayıcı İndirme Ayarlarının Yapılandırılması**

Tarayıcıyı başlatırken indirme işlemleri için ayarları yapılandırabilirsiniz. Örneğin, Chrome veya Firefox için indirme klasörünü belirlemek mümkündür.

from selenium import webdriver

# Chrome için indirme klasörünü belirleme

options = webdriver.ChromeOptions()

prefs = {

"download.default\_directory": "/path/to/download/directory", # İndirilecek dosya yolu

"download.prompt\_for\_download": False, # İndirme penceresi açılmasın

"download.directory\_upgrade": True,

"safebrowsing.enabled": True # Güvenli gözatma uyarılarını devre dışı bırak

}

options.add\_experimental\_option("prefs", prefs)

driver = webdriver.Chrome(options=options)

driver.get("URL")

**b. İndirilen Dosyanın Varlığını ve İçeriğini Doğrulama**

İndirilen dosyanın gerçekten var olup olmadığını ve içeriğini doğrulamak için Python'un os veya pathlib kütüphanesini kullanabilirsiniz.

import os

import time

# İndirme işlemi tamamlanana kadar bekle

download\_path = "/path/to/download/directory"

file\_name = "downloaded\_file.txt"

# Dosyanın indirilip indirilmediğini kontrol etmek için bekleme

timeout = 10 # saniye

while timeout:

if file\_name in os.listdir(download\_path):

print("Dosya başarıyla indirildi.")

break

else:

time.sleep(1)

timeout -= 1

# Dosyanın içeriğini kontrol etme

with open(os.path.join(download\_path, file\_name), 'r') as file:

content = file.read()

print("Dosya içeriği:", content)

# Importing necessary libraries

from selenium import webdriver

from selenium.webdriver.common.by import By

import os

import time

# Define file paths

upload\_file\_path = "C:\\Users\\Lenovo\\PycharmProjects\\Yazilim\_Gercekleme\_ve\_Test\\Week\_7\\Odev\_8.txt"

download\_directory = "C:\\Users\\Lenovo\\PycharmProjects\\Yazilim\_Gercekleme\_ve\_Test\\Week\_7\\downloads"

# Configure Chrome options for download settings

chrome\_options = webdriver.ChromeOptions()

prefs = {

"download.default\_directory": download\_directory,

"download.prompt\_for\_download": False,

"download.directory\_upgrade": True,

"safebrowsing.enabled": True

}

chrome\_options.add\_experimental\_option("prefs", prefs)

# Initialize the WebDriver

driver = webdriver.Chrome(options=chrome\_options)

try:

# Navigate to the target page for file upload and download

driver.get("URL") # Replace 'URL' with the actual URL of the target page

# --- Step 1: File Upload using send\_keys() ---

file\_input = driver.find\_element(By.XPATH, "//input[@type='file']")

file\_input.send\_keys(upload\_file\_path)

# Add any necessary validations for upload success

# --- Step 2: File Download Setup ---

# Assuming there is an element to trigger the file download

download\_button = driver.find\_element(By.XPATH, "//button[@id='download']")

download\_button.click()

# Verify file download

downloaded\_file = os.path.join(download\_directory, "Odev\_8.txt") # Adjust file name if needed

timeout = 10

while timeout > 0:

if os.path.exists(downloaded\_file):

print("File successfully downloaded.")

break

time.sleep(1)

timeout -= 1

# Optional: Check the content of the downloaded file

with open(downloaded\_file, 'r') as file:

content = file.read()

print("Downloaded file content:", content)

finally:

driver.quit()