Automatización de Navegadores Web con Selenium

Selenium es una herramienta poderosa para automatizar navegadores web. Permite controlar programáticamente un navegador para realizar acciones como navegar a páginas web, rellenar formularios, hacer clic en botones y extraer datos. Esto puede ser útil para una variedad de tareas, incluyendo web scraping, pruebas de aplicaciones web y automatización de tareas repetitivas.

Aquí hay un ejemplo básico de cómo usar Selenium con Python para raspar un blog de CSDN:

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException
import time

def scrape_csdn_blog(url):
    """
    Extrae un blog de CSDN y extrae todos los enlaces (etiquetas a) del código fuente de la página usando Sel
    filtrando los enlaces que comienzan con "https://blog.csdn.net/lzw_java/article".

    Args:
        url (str): La URL del blog de CSDN.
    """
    try:
        # Configurar las opciones de Chrome para la navegación sin cabeza
        chrome_options = Options()
        chrome_options.add_argument("--headless")  # Ejecutar Chrome en modo sin cabeza
        chrome_options.add_argument("--disable-gpu")  # Deshabilitar la aceleración de GPU (recomendado para s
        chrome_options.add_argument("--no-sandbox")  # Omitir el modelo de seguridad del sistema operativo
        chrome_options.add_argument("--disable-dev-shm-usage")  # Superar problemas de recursos limitados

        # Inicializar el controlador de Chrome
        driver = webdriver.Chrome(options=chrome_options)

        # Cargar la página web
        driver.get(url)

        # Encontrar todos los elementos de la etiqueta 'a'
        links = driver.find_elements(By.TAG_NAME, 'a')

        if not links:
```

```
            print("No se encontraron enlaces en la página.")
            driver.quit()
            return

        for link in links:
            try:
                href = link.get_attribute('href')
                if href and href.startswith("https://blog.csdn.net/lzw_java/article"):
                    text = link.text.strip()

                    print(f"Texto: {text}")
                    print(f"URL: {href}")
                    print("-" * 20)

            except Exception as e:
                print(f"Error extrayendo enlace: {e}")
                continue

    except Exception as e:
        print(f"Ocurrió un error: {e}")
    finally:
        # Cerrar el navegador
        if 'driver' in locals():
            driver.quit()

if __name__ == "__main__":
    blog_url = "https://blog.csdn.net/lzw_java?type=blog"  # Reemplazar con la URL real
    scrape_csdn_blog(blog_url)
```