Automatisation des navigateurs web avec Selenium

Selenium est un outil puissant pour automatiser les navigateurs web. Il permet de contrôler par programme un navigateur pour effectuer des actions telles que naviguer vers des pages web, remplir des formulaires, cliquer sur des boutons et extraire des données. Cela peut être utile pour diverses tâches, notamment le web scraping, les tests d'applications web et l'automatisation de tâches répétitives.

Voici un exemple basique d'utilisation de Selenium avec Python pour scraper un blog 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):
    """
    Scrape un



```python
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):
    """
    Scrape un blog CSDN et extrait tous les liens (balises a) de la source de la page à l'aide de Selenium,
    en filtrant les liens qui commencent par "https://blog.csdn.net/lzw_java/article".

    Args:
        url (str): L'URL du blog CSDN.
    """
    try:
        # Configuration des options Chrome pour la navigation sans tête
        chrome_options = Options()
        chrome_options.add_argument("--headless")  # Exécuter Chrome en mode sans tête
        chrome_options.add_argument("--disable-gpu")  # Désactiver l'accélération GPU (recommandé pour le mod
        chrome_options.add_argument("--no-sandbox")  # Contourner le modèle de sécurité du système d'exploita
```
```

```
        chrome_options.add_argument("--disable-dev-shm-usage")  # Surmonter les problèmes de ressources limit

        # Initialisation du pilote Chrome
        driver = webdriver.Chrome(options=chrome_options)

        # Chargement de la page web
        driver.get(url)

        # Trouver tous les éléments de balise 'a'
        links = driver.find_elements(By.TAG_NAME, 'a')

        if not links:
            print("Aucun lien trouvé sur la page.")
            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"Texte : {text}")
                    print(f"URL : {href}")
                    print("-" * 20)

            except Exception as e:
                print(f"Erreur lors de l'extraction du lien : {e}")
                continue

    except Exception as e:
        print(f"Une erreur s'est produite : {e}")
    finally:
        # Fermeture du navigateur
        if 'driver' in locals():
            driver.quit()

if __name__ == "__main__":
    blog_url = "https://blog.csdn.net/lzw_java?type=blog"  # Remplacer par l'URL réelle
    scrape_csdn_blog(blog_url)
```