

Automatisation de Connexion : Cas de Test Selenium WD, Playwright, Robot Framework pour l'Authentification Utilisateur



Exemple de Développement de Cas de Test avec Selenium WebDriver (Java) à partir d'une User Story :

User Story : En tant qu'utilisateur, je veux pouvoir me connecter à mon compte sur une application web de gestion des tâches avec des identifiants valides.

Cas de Test avec Selenium WebDriver (Java) :

Nom du Cas de Test : Connexion Utilisateur avec Identifiants Valides

Préconditions :

1. L'application web est accessible.
2. Les identifiants de l'utilisateur sont valides.

Étapes :

1. Ouvrir le navigateur et accéder à l'URL de l'application.
2. Sur la page d'accueil, saisir l'identifiant de l'utilisateur dans le champ "Nom d'utilisateur".
3. Saisir le mot de passe de l'utilisateur dans le champ "Mot de passe".
4. Cliquer sur le bouton "Se connecter".

```
1 import org.openqa.selenium.By;
2 import org.openqa.selenium.WebDriver;
3 import org.openqa.selenium.WebElement;
4 import org.openqa.selenium.chrome.ChromeDriver;
5
6 public class ConnexionUtilisateurTest {
7
8     public static void main(String[] args) {
9
10         WebDriver driver = new ChromeDriver();
11
12         driver.get("https://url");
13
14         WebElement usernameField = driver.findElement(By.name("username"));
15         WebElement passwordField = driver.findElement(By.name("password"));
16
17         usernameField.sendKeys("nom-utilisateur");
18         passwordField.sendKeys("mot-de-passe");
19
20         WebElement loginButton = driver.findElement(By.cssSelector("button[type=submit]"));
21         loginButton.click();
22
23         WebElement loggedInUser = driver.findElement(By.cssSelector(".user-name"));
24
25         if (loggedInUser.getText().equals("Nom de l'utilisateur connecté")) {
26             System.out.println("Test réussi : L'utilisateur est connecté avec succès.");
27         } else {
28             System.out.println("Test échoué : Problème de connexion.");
29         }
30
31         driver.quit();
32     }
33 }
```

Attendus :

1. L'utilisateur est redirigé vers la page d'accueil de l'application après la connexion.
2. La barre de navigation affiche le nom de l'utilisateur connecté.
3. Aucun message d'erreur n'est affiché.

Postconditions :

1. L'utilisateur est connecté à son compte.

Éléments de Test :

1. Vérifier le comportement en cas de saisie incorrecte d'identifiants.
2. Tester la connexion avec différents navigateurs.

Notes :

- S'assurer que la page de connexion est sécurisée (utilisation de HTTPS).

- Vérifier la gestion des sessions et la déconnexion.

Environnement :

- Système d'exploitation : Windows 10
- Navigateur : Google Chrome, Mozilla Firefox

Exemple de Développement de Cas de Test avec Playwright (JavaScript) à partir d'une User Story :**Cas de Test avec Playwright (JavaScript) :**

Nom du Cas de Test : Connexion Utilisateur avec Identifiants Valides

Préconditions :

1. L'application web est accessible.
2. Les identifiants de l'utilisateur sont valides.

Étapes :

```
const { chromium } = require('playwright');

(async () => {
  const browser = await chromium.launch();
  const page = await browser.newPage();

  await page.goto('https://url');

  await page.fill('input[name=username]', 'nom-utilisateur');
  await page.fill('input[name=password]', 'mot-de-passe');

  await page.click('button[type=submit]');

  await page.waitForNavigation();
  const loggedInUser = await page.textContent('.user-name');

  await browser.close();

  expect(loggedInUser).toBe('Nom de l\'utilisateur connecté');
})();
```

Attendus :

1. L'utilisateur est redirigé vers la page d'accueil de l'application après la connexion.

2. La barre de navigation affiche le nom de l'utilisateur connecté.
3. Aucun message d'erreur n'est affiché.

Postconditions :

1. L'utilisateur est connecté à son compte.

Éléments de Test :

1. Vérifier le comportement en cas de saisie incorrecte d'identifiants.
2. Tester la connexion avec différents navigateurs.

Notes :

- Assurez-vous d'installer Playwright et de configurer l'environnement.
- Vérifiez la stabilité des sélecteurs utilisés pour l'interaction avec les éléments de la page.

Environnement :

- Système d'exploitation : Windows 10
- Navigateur : Google Chrome

Exemple de Développement de Cas de Test avec Robot Framework (Python) à partir d'une User Story :**1. Settings Section :**

- **Library SeleniumLibrary** : Indique que nous utilisons la bibliothèque SeleniumLibrary pour l'automatisation des tests avec Robot Framework.

2. Variables Section :

- **\${BROWSER}** : Définit le navigateur que nous allons utiliser (dans cet exemple, Chrome).
- **\${URL}** : L'URL de l'application que nous allons tester.
- **\${USERNAME}** et **\${PASSWORD}** : Les identifiants de l'utilisateur.

3. Test Cases Section :

- **Connexion Utilisateur avec Identifiants Valides** : Nom du cas de test.
 - **Open Browser** : Ouvre le navigateur spécifié avec l'URL spécifiée.
 - **Input Text** : Saisit le nom d'utilisateur et le mot de passe dans les champs correspondants.
 - **Click Button** : Clique sur le bouton de connexion.

- **Page Should Contain** : Vérifie si la page contient le texte "Nom de l'utilisateur connecté".
- **[Teardown] Close Browser** : Ferme le navigateur après l'exécution du cas de test.

4. Exécution :

- Installez Robot Framework et la bibliothèque SeleniumLibrary.
- Exécutez le script en utilisant la commande `robot nom-du-fichier.robot` dans le terminal.

```
1 *** Settings ***
2 Library SeleniumLibrary
3
4 *** Variables ***
5 ${BROWSER} chrome
6 ${URL} https://url
7 ${USERNAME} nom-utilisateur
8 ${PASSWORD} mot-de-passe
9
10 *** Test Cases ***
11 Connexion Utilisateur avec Identifiants Valides
12     Open Browser ${URL} ${BROWSER}
13     Input Text name=username ${USERNAME}
14     Input Text name=password ${PASSWORD}
15     Click Button css:button[type=submit]
16     Page Should Contain Nom de l utilisateur connecté
17     [Teardown] Close Browser
```