# Installer Playwright avec JavaScript et Cucumber – Étapes et Exemple de Développement d'un Cas de Test

## Introduction

Playwright est un outil puissant de test end-to-end, développé par Microsoft, qui permet de tester les applications web modernes avec différentes options de navigateurs. En l'intégrant avec Cucumber, vous pouvez bénéficier de la structure des scénarios BDD (Behavior Driven Development) pour une meilleure lisibilité et compréhension de vos cas de test.

## Étape 1 : Prérequis

Avant de commencer, assurez-vous d'avoir les éléments suivants installés sur votre machine :

- **Node.js** : Playwright fonctionne avec Node.js. Vous pouvez le télécharger et l'installer depuis le site officiel de Node.js.
- **NPM** (Node Package Manager) : NPM est fourni avec Node.js.
- **VS Code** ou un autre éditeur de code pour écrire et exécuter vos scripts.

## Étape 2 : Créer un nouveau projet

1. **Initialiser un projet Node.js :** Créez un nouveau dossier pour votre projet, puis ouvrez votre terminal et exécutez les commandes suivantes :

```
mkdir playwright-cucumber-demo
cd playwright-cucumber-demo
npm init -y
```

Cela créera un fichier `package.json` de base pour votre projet.

## Étape 3 : Installer les dépendances

1. **Installer Playwright :** Installez Playwright en exécutant la commande suivante dans votre terminal :

```
npm install playwright
```

Cela installera les navigateurs par défaut (Chromium, Firefox et WebKit) pour les tests.

2. **Installer Cucumber.js et d'autres dépendances :**

Installez Cucumber.js et d'autres outils nécessaires pour le support des tests BDD :

```
npm install @cucumber/cucumber chai
```

3. **Configurer Babel pour la compatibilité ES6 :**

   Installez Babel pour utiliser la syntaxe moderne de JavaScript avec Cucumber :

```
npm install @babel/core @babel/preset-env @babel/register
```

   Créez un fichier `.babelrc` à la racine de votre projet et ajoutez la configuration suivante :

```
{
  "presets": ["@babel/preset-env"]
}
```

4. **Installer Playwright-Cucumber :**

   Installez également `playwright-cucumber`, une intégration de Playwright avec Cucumber :

```
npm install playwright-cucumber
```

## Étape 4 : Configuration du projet

1. **Créer la structure de fichiers :**

   Créez les dossiers suivants dans votre projet pour organiser vos fichiers de test :

```
└── features
    └── step-definitions
    └── support
```

2. **Configurer Cucumber.js :**

   Créez un fichier `cucumber.js` à la racine de votre projet pour configurer Cucumber :

```
module.exports = {
  default: `--require-module @babel/register --require features/step-definitions/*.js --format progress`
};
```

3. **Initialiser Playwright dans les fichiers de support :**

Dans le dossier `support`, créez un fichier `world.js` pour initialiser Playwright :

```js
const { setWorldConstructor } = require('@cucumber/cucumber');
const { chromium } = require('playwright');

class CustomWorld {
  async launchBrowser() {
    this.browser = await chromium.launch({ headless: false });
    this.context = await this.browser.newContext();
    this.page = await this.context.newPage();
  }

  async closeBrowser() {
    await this.browser.close();
  }
}

setWorldConstructor(CustomWorld);
```

## Étape 5 : Rédiger le Scénario de Test

Dans le dossier `features`, créez un fichier `search.feature` pour décrire un scénario de test avec Cucumber :

```
Feature: Search on Google

  Scenario: User can search for Playwright
    Given the user is on the Google homepage
    When the user searches for "Playwright"
    Then the user should see results related to "Playwright"
```

## Étape 6 : Définir les Étapes

Dans le dossier `step-definitions`, créez un fichier `searchSteps.js` et définissez les étapes pour ce scénario :

```
const { Given, When, Then } = require('@cucumber/cucumber');
const { expect } = require('chai');

Given('the user is on the Google homepage', async function () {
  await this.launchBrowser();
  await this.page.goto('https://www.google.com');
});

When('the user searches for {string}', async function (searchTerm) {
  await this.page.fill('input[name="q"]', searchTerm);
  await this.page.press('input[name="q"]', 'Enter');
});

Then('the user should see results related to {string}', async function (searchTerm) {
  const text = await this.page.textContent('body');
  expect(text).to.include(searchTerm);
  await this.closeBrowser();
});
```

## Étape 7 : Exécuter le Test

Maintenant, vous pouvez exécuter votre test en utilisant la commande suivante :

```
npx cucumber-js
```

Cela lancera Playwright et exécutera votre scénario de test Cucumber.