

Guide sur Cucumber

Hedi bensaid

Qu'est-ce que Cucumber ?

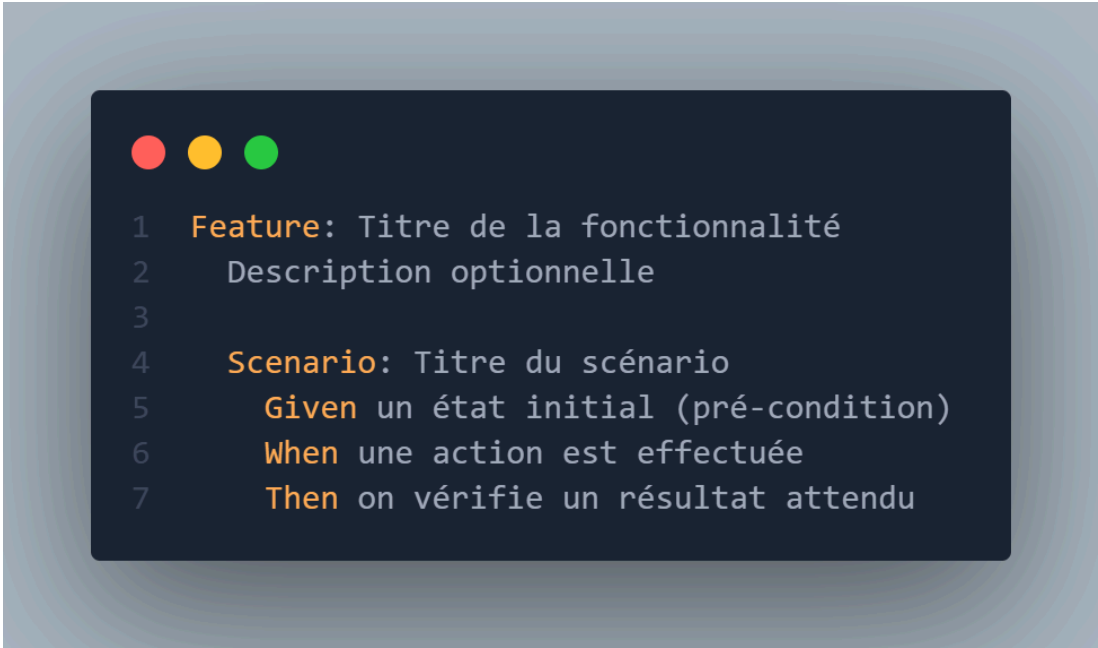
Cucumber est un outil **BDD (Behavior Driven Development)** qui permet d'écrire des scénarios de test en langage naturel (**Gherkin**).

Cela facilite la collaboration entre les métiers (Product Owner, QA, Dev).

1. La syntaxe Gherkin

La syntaxe Gherkin est simple et lisible, permettant de décrire les fonctionnalités et scénarios de manière claire et concise.

Exemple de syntaxe

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains Gherkin syntax examples with line numbers 1 through 7.

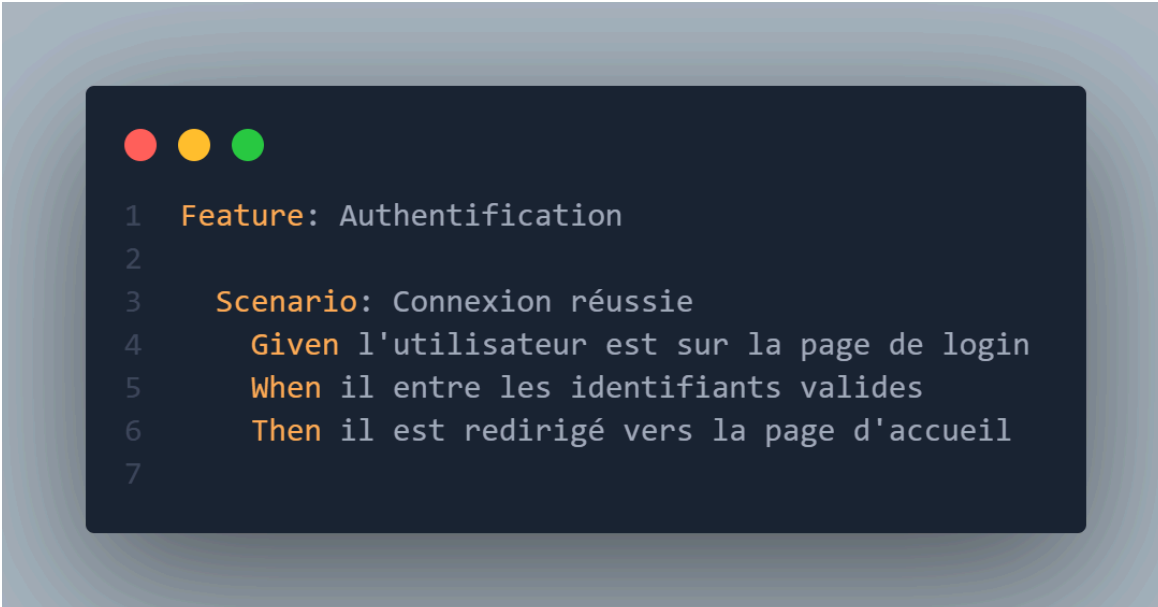
```
1 Feature: Titre de la fonctionnalité
2   Description optionnelle
3
4   Scenario: Titre du scénario
5     Given un état initial (pré-condition)
6     When une action est effectuée
7     Then on vérifie un résultat attendu
```

2. Les mots-clés principaux et leur objectif

Voici un tableau des mots-clés utilisés dans Gherkin et leur usage :

Mot-clé	Usage
Feature	Décrit la fonctionnalité à tester
Scenario	Un cas de test précis
Given	Décrit le contexte / état initial
When	Décrit l'action principale à effectuer
Then	Décrit le résultat attendu (assertion)
And / But	Pour chaîner plusieurs Given / When / Then
Background	Préconditions communes à tous les scénarios d'une feature


Exemple simple



```
1 Feature: Authentification
2
3 Scenario: Connexion réussie
4   Given l'utilisateur est sur la page de login
5   When il entre les identifiants valides
6   Then il est redirigé vers la page d'accueil
7
```


3. Comment passer un paramètre ?

Il est possible de passer des paramètres dans les steps en utilisant des expressions. Par exemple :



```
1 Scenario: Rechercher un produit
2   Given l'utilisateur est sur la page d'accueil
3   When il recherche le produit "iPhone"
4   Then la liste des résultats contient "iPhone"
5
```

Dans le fichier de définition (ex : steps.js pour Cypress Cucumber ou steps.rb pour Ruby) :



```
1 When('il recherche le produit {string}', (produit) => {
2   cy.get('input[aria-label="Search"]').type(produit);
3   cy.get('button[type="submit"]').click();
4 });
```

4. Comment tester avec un tableau (DataTable) ?

Les **DataTables** sont utilisées pour les tests multi-variables et sont définies avec le mot-clé |.

Exemple avec un tableau

```

1
2 Scenario: Vérifier plusieurs utilisateurs
3   Given les utilisateurs suivants existent :
4       | nom      | email                |
5       | Mohamed  | mohamed@test.com    |
6       | Ali      | ali@test.com         |
7       | Hedi     | hedi@test.com       |
8   When je liste les utilisateurs
9   Then je devrais voir "Mohamed", "Ali", "Hedi"
10

```

Dans le step :

```

1 Given('les utilisateurs suivants existent :', (dataTable) => {
2   dataTable.hashes().forEach(user => {
3     cy.request('POST', '/api/users', {
4       name: user.nom,
5       email: user.email
6     });
7   });
8 });
9

```

dataTable.hashes() convertit le tableau Gherkin en un tableau d'objets JS :

```
1  [  
2    { nom: 'Alice', email: 'alice@test.com' },  
3    { nom: 'Bob', email: 'bob@test.com' },  
4    { nom: 'Charlie', email: 'charlie@test.com' }  
5  ]
```

5. Autres syntaxes utiles

✓ Background pour un Given commun à tous les scénarios :

```
1  Feature: Profil utilisateur  
2  
3    Background:  
4      Given l'utilisateur est connecté  
5  
6    Scenario: Modifier son profil  
7      When il change son nom  
8      Then son profil est mis à jour  
9  
10   Scenario: Supprimer son compte  
11     When il clique sur supprimer  
12     Then le compte est supprimé
```

 Conclusion : Résumé rapide

Syntaxe	Rôle
Feature	Décrit la fonctionnalité
Scenario	Décrit un cas de test
Given	Contexte initial
When	Action réalisée
Then	Résultat attendu
And / But	Chaîner les steps
Background	Given commun à tous les scénarios