Réexécuter Uniquement les Tests Échoués avec Selenium, Cucumber, TestNG dans une Pipeline Jenkins

Lorsqu'un test échoue dans une suite de tests automatisés, il peut être utile de réexécuter uniquement les tests échoués pour vérifier s'il s'agissait d'une défaillance temporaire ou d'un problème persistant. Dans cet article, nous allons explorer une solution complète pour gérer la réexécution des tests échoués dans une configuration utilisant Selenium, Cucumber, TestNG et Jenkins.

1. Introduction à la gestion des tests échoués

Dans un environnement de test continu, les tests peuvent échouer pour diverses raisons :

- Des délais d'attente trop courts.
- Des dépendances instables.
- Des problèmes réseau temporaires.

Au lieu de relancer l'ensemble de la suite de tests, il est plus efficace de cibler uniquement les tests échoués.

2. Configuration de TestNG pour réexécuter les tests échoués

TestNG propose une solution native pour réexécuter les tests échoués via un fichier XML de résultats.

Étape 1 : Activer le Listener TestNG

Ajoutez un listener à votre fichier `testng.xml` pour capturer les tests échoués :

```xml
<suite name="Test Suite" verbose="1" parallel="false">
    <listeners>
        <listener class-name="org.testng.reporters.FailedReporter" />
    </listeners>
    <test name="Regression Test">
        <classes>
            <class name="com.project.tests.MyTest" />
        </classes>
    </test>
</suite>
```

Étape 2 : Générer un fichier de tests échoués

Après l'exécution initiale, TestNG crée un fichier nommé `testng-failed.xml` qui contient uniquement les tests ayant échoué.

Étape 3 : Réexécuter les tests échoués

Dans votre pipeline Jenkins ou localement, vous pouvez exécuter la commande suivante :

```
java -cp "path/to/testng.jar:path/to/your/tests" org.testng.TestNG test-output/testng-failed.xml
```

3. Utiliser Cucumber avec un Plugin de Réexécution

Pour Cucumber, la réexécution des tests échoués est facilitée par une option intégrée.

Étape 1 : Ajouter le Plugin de Réexécution

Configurez votre runner avec le plugin `rerun`:

```
@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/resources/features",
    glue = "com.project.stepdefinitions",
    plugin = {
        "pretty",
        "html:target/cucumber-reports",
        "rerun:target/failed-scenarios.txt"
    }
)
public class TestRunner {}
```

Ce plugin génère un fichier `failed-scenarios.txt` contenant les chemins des scénarios échoués.

Étape 2 : Réexécuter les Scénarios Échoués

Ajoutez une nouvelle tâche dans votre pipeline Jenkins pour réexécuter uniquement ces tests :

```
mvn test -Dcucumber.features=@target/failed-scenarios.txt
```

4. Configurer une Pipeline Jenkins pour Gérer les Tests Échoués

Voici un exemple de pipeline Jenkins pour exécuter des tests et réexécuter ceux qui ont échoué.

Étape 1 : Définir le Jenkinsfile

```
pipeline {
    agent any
    stages {
        stage('Checkout Code') {
            steps {
                git url: 'https://github.com/your-repo/project.git', branch: 'main'
            }
        }
        stage('Run Tests') {
            steps {
                script {
                    try {
                        sh 'mvn clean test'
                    } catch (Exception e) {
                        currentBuild.result = 'UNSTABLE'
                    }
                }
            }
        }
        stage('Re-run Failed Tests') {
            when {
                expression { fileExists('target/failed-scenarios.txt') }
            }
            steps {
                sh 'mvn test -Dcucumber.features=@target/failed-scenarios.txt'
            }
        }
    }
    post {
        always {
            junit 'target/surefire-reports/*.xml'
        }
    }
}
```

Étape 2 : Analyser les Résultats

Ajoutez le plugin **JUnit** dans Jenkins pour visualiser les résultats des tests.

5. Exemple Réel

Supposons que vous avez un scénario Cucumber échouant :

```
Feature: Login Functionality

  Scenario: Login with invalid credentials
    Given I am on the login page
    When I enter "invalid_user" and "wrong_password"
    Then I should see an "Invalid credentials" message
```

Après l'exécution initiale, `target/failed-scenarios.txt` contient :

```
src/test/resources/features/Login.feature:3
```

En relançant uniquement ce fichier, vous économisez du temps et des ressources.

6. Conseils pour un Pipeline Stable

- **Gérer les dépendances** : Assurez-vous que tous les services externes nécessaires sont opérationnels.
- **Utiliser des délais d'attente dynamiques** : Évitez les échecs causés par des délais d'attente trop courts.
- **Intégrer des notifications** : Configurez des alertes pour informer l'équipe des tests récurrents échoués.