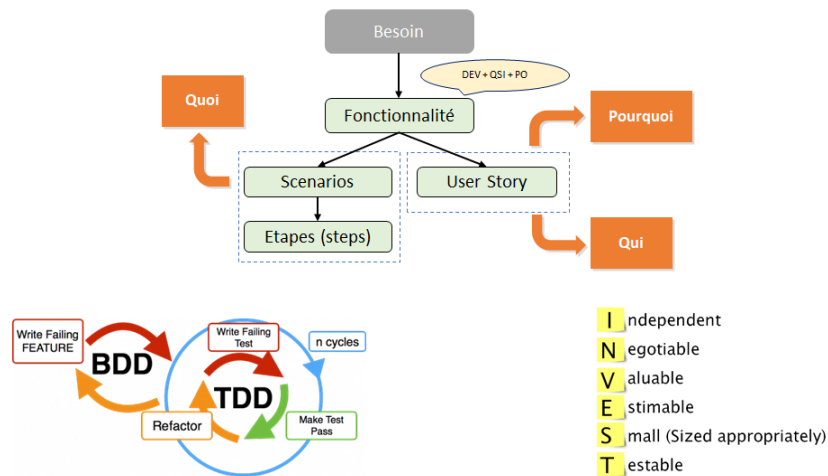


Tests Unitaires : TDD, BDD

Créée par JELLAD Abir, dernière modification par HANNOT THOMAS le 10. déc. 2019

Behavior Driven Development / Test Driven Development



Cucumber-Eclipse est un plug-in Eclipse qui permet d'écrire et d'exécuter les scénarios Cucumber directement dans l'IDE facilitant ainsi leur automatisation.

1- Configuration du pom.xml pour Cucumber:

```
<dependencies>
  <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>3.0.2</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-junit</artifactId>
    <version>3.0.2</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

2- Ajout des ressources .feature

Créez un nouveau Source Folder que vous appellerez `src/test/resources`. Créez un fichier `cocktail.feature` dans un package `dojo` de `src/test/resources`.

Le plug-in Cucumber-Eclipse reconnaît l'extension `.feature` et vous remplit automatiquement ce fichier en vous montrant différentes manières d'écrire un scénario.

Effacez le contenu de ce fichier, ne gardez que la première ligne dans laquelle vous pouvez ajouter vos noms et prénoms.

Exemple 1

```
Feature: Cocktail Ordering
  As Romeo, I want to offer a drink to Juliette so that we can discuss together (and maybe more).

Scenario: Creating an empty order
  Given Romeo who wants to buy a drink
  When an order is declared for Juliette
  Then there is no cocktail in the order
```

Exemple 2

```
Feature: Gestion des Virements
  Dans le but de pouvoir gérer mes comptes
  En tant que client banque
```

```
Je souhaite pouvoir effectuer des virements entre mes comptes
```

```
Scenario: Virement simple
```

```
Given j'ai un compte cheque avec un solde de 500€
```

```
And j'ai un compte épargne avec un solde de 0€
```

```
When j'effectue un virement de 100€ du compte cheque vers le compte épargne
```

```
Then le solde du compte cheque est 400€
```

```
And le solde du compte épargne est 100€
```

```
And le virement est confirmé
```

Dans Cucumber, une story est composée de scénarios et chaque scénario est composé d'étapes. La story est (d)écrite en Gherkin dans un fichier .feature dans lequel on trouve :

le titre de la story introduit par le mot clé Feature ; Feature: <Titre de la fonctionnalité>

Un descriptif (optionnel) (qui ne sera pas interprété par Cucumber) permettant par exemple, de résumer la story à l'aide du template As..., I want to..., so that... et/ou de noter toute autre information utile à connaître ;

les scénarios de la story :

- Chaque scénario est introduit par le mot clé Scenario. Ce mot clé peut être suivi ou non d'un titre qui décrit explicitement le critère d'acceptation de la story associée à ce scénario,
- Un scénario étant un exemple concret qui illustre une règle métier, il est composé de plusieurs étapes. Les différentes étapes d'un scénario sont décrites à partir des trois principaux mots clés : Given, When et Then suivant la place et le rôle qu'elles occupent dans le scénario :
 - Given décrit les conditions initiales du scénario, c.-à-d. le contexte dans lequel va se dérouler le scénario,
 - When décrit une action effectuée par un utilisateur, c.-à-d. un événement qui va réellement déclencher le scénario,
 - Then décrit le comportement attendu, ce qui devrait se produire lorsque les conditions initiales sont remplies et l'action est effectuée.

3- Configuration du lanceur de test

Implémenter un fichier **RunCucumberTest.java** dans un package dojo de *src/test/java*

de manière qu'il devienne le lanceur des tests Cucumber :

```
package dojo;

import cucumber.api.junit.Cucumber;
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
public class RunCucumberTest {
}
```

- L'annotation `@RunWith(Cucumber.class)` permet d'exécuter les scénarios Cucumber dans JUnit à l'aide du JUnit Runner.
- Le lanceur de test fait le mapping entre les étapes des scénarios écrites de manière naturelle (fichier .feature) et les méthodes Java implémentant ces étapes. Les étapes Java sont appelées des steps et sont définies grâce à des annotations spécifiques : `@Given`, `@When`, `@Then`.

4- Implémentation STEP

Exemple 1

```
package dojo;

import cucumber.api.PendingException;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;

public class CocktailSteps {

    @Given("Romeo who wants to buy a drink")
    public void romeo_who_wants_to_buy_a_drink() {
        order = new Order();
        order.declareOwner("Romeo");
    }

    @When("an order is declared for Juliette")
    public void an_order_is_declared_for_Juliette() {
        order.declareTarget("Juliette");
    }

    @Then("there is no cocktail in the order")
    public void there_is_no_cocktail_in_the_order() {
        List<String> cocktails = order.getCocktails();
        assertEquals(0, cocktails.size());
    }
}
```

Exemple 2

```
public class GestionVirementScenario {

    private Compte compteDebitueur;
```

```
private Compte compteCrediteur;
private String result;

@Given("un compte chèque avec un solde de {int} €")
public void un_compte_chèque_avec_un_solde_de_€(Integer solde) {
    this.compteDebiteur = new Compte();
    this.compteDebiteur.setSolde(solde);
}

@Given("un compte épargne avec un solde de {int} €")
public void un_compte_épargne_avec_un_solde_de_€(Integer solde) {
    this.compteCrediteur = new Compte();
    this.compteCrediteur.setSolde(solde);
}

@When("j effectue un virement de {int} €")
public void j_effectue_un_virement_de_€(int montant) {
    result = compteDebiteur.virement(montant);
}

@Then("j obtiens un message d erreur {string}")
public void j_obtiens_un_message_d_erreur(String msgErr) {
    assertEquals(msgErr, result);
}
}
```

5- Exemple de tutos:

- <https://isabelle-blasquez.developpez.com/tutoriels/java/cucumber-test-bdd/>
- https://github.com/iblasquez/tuto_bdd_cucumber

Vidéo Hands ON :

- <http://webtv.axa-fr.intraxa/app/video/6a76eec7-785d-5687-a8f6-d6d8e443b691>

Aucune étiquette

Community & Marketplace News By using this Confluence Instance you agree to this Terms & Conditions and all applicable laws and regulations. This documentation is provided by Cloud Foundations - © 2019 by AXA Services