

## **Orinoco**

Orinoco est une enseigne marchande sur internet qui veut spécialiser chacune de ses boutiques dans un domaine particulier. Il y aurait entre autre Orinoco Jouet, Orinoco Ameublement ou encore Orinoco Appareil électronique.

Le plan de test est découpé en deux parties.

La première, suit la technique dite BDD : Behavior Driven Developpement. Nous utiliseront le logiciel Cucumber.

La deuxième utilise les TDD : Test Driven Developpement et utilisera le logiciel Jasmine.

### **I) Behavior Driven Developpement avec Cucumber**

Comme son nom l'indique, ces tests guident le développement par l'approche comportementale de l'application.

Pour cela cucumber nous propose de rédiger une feature composée d'un ou plusieurs scénarii.

#### **A) La feature**

Dans le dossier du projet nous avons un dossier BDD, qui héberge cucumber.

Notre fichier .feature s'intitule boutique\_specialisee , pour suivre la logique de l'enseigne. Notre organisation est un moyen de faire des tests qui , non seulement contrôleront le comportement de l'application. Mais qui garantiront qu'elle est en accord avec les desiderata du client.

##### **1) Boutique Spécialisée**

La feature s'enonce donc ainsi :

*En tant que client , je souhaite pouvoir sélectionner un produit dans la liste voulu, et passer commande.*

Pour ce faire nous avons découpé la feature en deux scenarii :

##### **a) Le premier scénario**

*Je vais dans la boutique ours en peluche pour en acheter un.*

Le comportement est ainsi décrit :

Lorsque (Given) je clique sur la boutique ours en peluche

Alors (Then) la page affiche les données du serveur

##### **b) Le deuxième scénario**

*Sur la page des ours en peluche.*

Le comportement est ainsi décrit :

Lorsque (Given) choisi un produit, une page ouvre

Quand (When) j'ai personnalisé les options, je l'ajoute à mon panier

Alors (Then) je rempli correctement le formulaire pour passer commande

Et (And) un message un message me confirme mon achat

## **II) Test Driven developpement avec Jasmine**

Avec jasmine nous contrôlerons que les fonctionnalités créées en suivant la BDD fonctionnent correctement.

### **A) La spec**

Dans le dossier du projet on trouve le dossier Jasmine. Le fichier PelucheSpec contrôle l'existence de toutes les fonctions nécessaire à l'application.

#### **1) Les fonctions**

Énoncées littéralement comme suit :

- Écouteur d'évènement sur le lien sélectionné
- La réponse est parsée
- Les produits sont affichés dynamiquement sur la page
- Clic sur un produit
- Affichage du produit choisi
- Personnalisation des options
- Ajout au panier
- Contrôle de la validité des données
- Envoi de la commande
- Confirmation de commande est parsée
- Affichage de orderID

## **II) Le plan de test**

La combinaison de ces deux techniques nous offre donc un plan de test qui couvre 100 % du code de cette MVP, que vous trouverez ci dessous. En bleu le comportement et en orange les fonctions associées.

*Feature: boutique\_specialisee*

*En tant que client , je souhaite pouvoir sélectionner un produit dans la liste voulue, et passer commande.*

*Scenario: Je vais dans la boutique ours en peluche pour en acheter un.*

*Given je clique sur la boutique meuble en chêne*

*Écouteur d'évènement sur le lien sélectionné*

*Then la page affiche les données du serveur*

*La réponse est parsée*

*Les produits sont affichés dynamiquement sur la page*

*Scenario: Sur la page des ours en peluche*

*Given je choisi un produit, une page s'ouvre*

*clic sur un produit*

*Affichage du produit choisi*

*When j'ai personnalisé les options, je l'ajoute à mon panier*

*Personnalisation des options*

*Ajout au panier*

*Then je rempli correctement le formulaire pour passer commande*

*contrôle de la validité des données*

*Envoi de la commande*

*And un message me confirme mon achat*

*Confirmation de commande est parsée*

*affichage de orderID*

L'ensemble de ses test sont susceptible de produire automatiquement de la documentation.