



# **PROJET TA ROBOTFRAMEWORK**

---

**ISI GL 2025 - GROUPE 1**

**BOUKAKA CHANCELVIE  
PROF: M. MOUSSA WADE**

Institut Supérieur d'Informatique

---

## Lab N°1: Tests sur une base de données MongoDB

Cahier de Tests avec différents scénarios :

### Entité: Produit

Supposons l'entité **Produit** qui contient comme propriétés : **id, title, price, description, category, image**, nous allons effectuer une série de scénarios (passant et non passant dessus).

#### Ajout Produit

- Scénario Passant :

Requête POST avec tous les champs (propriétés) valides.

Résultat : Code HTTP 201 + corps de réponse avec l'ID généré.

- Scénario Non Passant N°1:

Requête POST avec un champ obligatoire non rempli (le titre par exemple).

Résultat : Code 400 ou un message d'erreur.

- Scénario Non Passant N°2:

Requête POST avec des données de type incorrect (le prix en chaîne de caractère, par exemple).

Résultat : Notification erreur de validation

#### Lecture Produit

- Scénario Passant :

Lire un produit existant par son ID.

Résultat : Code HTTP 200 avec les bonnes informations.

- Scénario Non Passant N°1 :

Lire un produit avec un ID qui n'existe pas.

Résultat : Code HTTP 404 Not Found.

- Scénario Non Passant N°2 :

Lire un produit avec un ID qui n'est pas valide (l'id en chaîne de caractères, par exemple).

Résultat : Code HTTP 400 Bad Request.

## Modification Produit

- **Scénario Passant :**

Modifier un produit existant avec des données valides.

Résultat : Code HTTP 200 OK + nouvelles valeurs.

- **Scénario Non Passant N°1 :**

Modifier un produit inexistant.

Résultat : Code HTTP 404.

- **Scénario Non Passant N°2 :**

Modifier un produit avec des champs invalides.

Résultat : Code HTTP 400 ou erreur de validation.

## Suppression Produit

- **Scénario passant :**

Supprimer un produit existant.

Résultat : Code HTTP 200 OK ou 204 No Content.

- **Scénario non passant N°1 :**

Supprimer un produit inexistant.

Résultat : Code HTTP 404.

- **Scénario non passant N°2 :**

Supprimer avec un ID malformé.

Résultat : Code HTTP 400.

## Entité: Carte

Supposons l'entité **Carte** qui contient comme propriétés : **id**, **userId**, **products** (qui est une liste de produits), nous allons effectuer une série de scénarios (passant et non passant dessus).

## Ajout Carte

- **Scénario Passant :**

Requête POST avec tous les champs (propriétés) valides.

Résultat : Code HTTP 201 + corps de réponse avec l'ID généré.

- **Scénario Non Passant N°1:**

Requête POST avec un champ obligatoire non rempli (le nom de l'utilisateur par exemple).

Résultat : Code 400 ou un message d'erreur.

- **Scénario Non Passant N°2:**

Requête POST avec des données de type incorrect (l'insertion d'un produit en chaîne de caractère et non sous forme d'un tableau de produits, par exemple).

Résultat : Notification erreur de validation

## Lecture Carte

- **Scénario Passant :**

Lire une carte existant par son ID.

Résultat : Code HTTP 200 avec les bonnes informations.

- **Scénario Non Passant N°1 :**

Lire une carte avec un ID qui n'existe pas.

Résultat : Code HTTP 404 Not Found.

- **Scénario Non Passant N°2 :**

Lire une carte avec un ID qui n'est pas valide (l'id en chaîne de caractères, par exemple).

Résultat : Code HTTP 400 Bad Request.

## Modification Carte

- **Scénario Passant :**

Modifier une carte existante avec des données valides.

Résultat : Code HTTP 200 OK + nouvelles valeurs.

- **Scénario Non Passant N°1 :**

Modifier une carte inexistante.

Résultat : Code HTTP 404.

- **Scénario Non Passant N°2 :**

Modifier une carte avec des champs invalides.

Résultat : Code HTTP 400 ou erreur de validation.

## Suppression Carte

- **Scénario passant :**

Supprimer une carte existante.

Résultat : Code HTTP 200 OK ou 204 No Content.

- **Scénario non passant N°1 :**

Supprimer une carte inexistante.

Résultat : Code HTTP 404.

- **Scénario non passant N°2 :**

Supprimer une carte avec un ID malformé.

Résultat : Code HTTP 400.

## Entité: Utilisateur

Supposons l'entité **Utilisateur** qui contient comme propriétés : **id, username, email, password**, nous allons effectuer une série de scénarios (passant et non passant dessus).

### Ajout Utilisateur

- **Scénario Passant :**

Requête POST avec tous les champs (propriétés) valides.

Résultat : Code HTTP 201 + corps de réponse avec l'ID généré.

- **Scénario Non Passant N°1:**

Requête POST avec un champ obligatoire non rempli (l'email de l'utilisateur par exemple).

Résultat : Code 400 ou un message d'erreur.

- **Scénario Non Passant N°2:**

Requête POST avec des données de type incorrect (l'email de l'utilisateur, par exemple).

Résultat : Notification erreur de validation

### Lecture Utilisateur

- **Scénario Passant :**

Lire un utilisateur existant par son ID.

Résultat : Code HTTP 200 avec les bonnes informations.

- **Scénario Non Passant N°1 :**

Lire un utilisateur avec un ID qui n'existe pas.

Résultat : Code HTTP 404 Not Found.

- **Scénario Non Passant N°2 :**

Lire un utilisateur avec un ID qui n'est pas valide (l'id en chaîne de caractères, par exemple).

Résultat : Code HTTP 400 Bad Request.

## **Modification Utilisateur**

- **Scénario Passant :**

Modifier un utilisateur existant avec des données valides.

Résultat : Code HTTP 200 OK + nouvelles valeurs.

- **Scénario Non Passant N°1 :**

Modifier un utilisateur inexistant.

Résultat : Code HTTP 404.

- **Scénario Non Passant N°2 :**

Modifier un utilisateur avec des champs invalides.

Résultat : Code HTTP 400 ou erreur de validation.

## **Suppression Utilisateur**

- **Scénario passant :**

Supprimer un utilisateur existant.

Résultat : Code HTTP 200 OK ou 204 No Content.

- **Scénario non passant N°1 :**

Supprimer un utilisateur inexistant.

Résultat : Code HTTP 404.

- **Scénario non passant N°2 :**

Supprimer un utilisateur avec un ID malformé.

Résultat : Code HTTP 400.

## Automatisation des tests avec Robot Framework.

Vous trouverez les fichiers de programme **.robot** concernés par ces test, joins au dossier du projet, dans le sous-dossier **Lab N°1**.

Faudra au préalable installer ces modules, pour le bon fonctionnement de ces programmes.

**pip install robotframework**

**pip install robotframework-requests**

Ci-dessous les captures d'écran faisant foi des différents tests :

## Test Produit

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests> robot crud_tests_produit.robot
=====
Crud Tests Produit
=====
Ajouter un produit - Scénario Passant                                .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Ajouter un produit - Scénario Passant                                | PASS |
-----
Ajouter un produit - Scénario Non Passant - Champ manquant          .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Ajouter un produit - Scénario Non Passant - Champ manquant          | FAIL |
200 != 400
-----
Ajouter un produit - Scénario Non Passant - Type invalide           .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Ajouter un produit - Scénario Non Passant - Type invalide           | FAIL |
200 != 400
-----
Lire un produit - Scénario Passant                                   .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Lire un produit - Scénario Passant                                   | PASS |
-----
Lire un produit - Scénario Non Passant - ID inexistant              C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised.
warnings.warn(
Lire un produit - Scénario Non Passant - ID inexistant              | FAIL |
200 != 404
-----
Lire un produit - Scénario Non Passant - ID invalide                C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised.
warnings.warn(
Lire un produit - Scénario Non Passant - ID invalide                | FAIL |
200 != 400
-----
Modifier un produit - Scénario Passant                               .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Modifier un produit - Scénario Passant                               | PASS |
-----
Modifier un produit - Scénario Non Passant - ID inexistant          .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Modifier un produit - Scénario Non Passant - ID inexistant          | FAIL |
200 != 404
-----
Modifier un produit - Scénario Non Passant - Données invalides     .C:\Users\bchan\AppData\Local\Temp\robot_20250924_150000\
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised
warnings.warn(
Modifier un produit - Scénario Non Passant - Données invalides     | FAIL |
200 != 400
-----
```



```
-----
Supprimer un produit - Scénario Passant                                C:\Users\bchan\AppData\
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised.
warnings.warn(
Supprimer un produit - Scénario Passant                                | PASS |
-----
Supprimer un produit - Scénario Non Passant - ID inexistant          C:\Users\bchan\AppData\
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised.
warnings.warn(
Supprimer un produit - Scénario Non Passant - ID inexistant          | FAIL |
200 != 404
-----
Supprimer un produit - Scénario Non Passant - ID invalide           C:\Users\bchan\AppData\
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advised.
warnings.warn(
Supprimer un produit - Scénario Non Passant - ID invalide           | FAIL |
HTTPError: 400 Client Error: Bad Request for url: https://fakestoreapi.com/products/abc
-----
Crud Tests Produit                                                  | FAIL |
12 tests, 4 passed, 8 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests> |
```

## Test Carte

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests> robot crud_tests_carte.robot
=====
Crud Tests Carte
=====
Ajout Carte - Scénario Passant ..C:\Users\bchan\AppData
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly adv
warnings.warn(
Ajout Carte - Scénario Passant | PASS |
-----
Ajout Carte - Champ manquant | FAIL |
AttributeError: 'str' object has no attribute 'items'
-----
Ajout Carte - Format produit incorrect | FAIL |
AttributeError: 'str' object has no attribute 'items'
-----
Lecture Carte - Scénario Passant C:\Users\bchan\AppData
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advis
warnings.warn(
[ WARN ] Keyword 'RequestsLibrary.To Json' is deprecated. Please use ${resp.json()} instea
Lecture Carte - Scénario Passant | PASS |
-----
Lecture Carte - ID inexistant C:\Users\bchan\AppData
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advis
warnings.warn(
Lecture Carte - ID inexistant | FAIL |
Url: https://fakestoreapi.com/carts/9999 Expected status: 200 != 404
-----
Lecture Carte - ID malformé C:\Users\bchan\AppData
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advis
warnings.warn(
Lecture Carte - ID malformé | FAIL |
HTTPError: 400 Client Error: Bad Request for url: https://fakestoreapi.com/carts/abc
-----
Modification Carte - Scénario Passant ..C:\Users\bchan\AppData
is being made to host 'fakestoreapi.com'. Adding certificate verification is strongly adv
warnings.warn(
Modification Carte - Scénario Passant | PASS |
-----
Modification Carte - Carte inexistant | FAIL |
AttributeError: 'str' object has no attribute 'items'
-----
Modification Carte - Données invalides | FAIL |
AttributeError: 'str' object has no attribute 'items'
-----
```

```
-----
Suppression Carte - Scénario Passant C:\Users\bchan\AppData
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advis
warnings.warn(
Suppression Carte - Scénario Passant | PASS |
-----
Suppression Carte - Carte inexistante C:\Users\bchan\AppData
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advis
warnings.warn(
Suppression Carte - Carte inexistante | FAIL |
Url: https://fakestoreapi.com/carts/9999 Expected status: 200 != 404
-----
Suppression Carte - ID malformé C:\Users\bchan\AppData
s being made to host 'fakestoreapi.com'. Adding certificate verification is strongly advis
warnings.warn(
Suppression Carte - ID malformé | FAIL |
HTTPError: 400 Client Error: Bad Request for url: https://fakestoreapi.com/carts/abc
-----
Crud Tests Carte | FAIL |
12 tests, 4 passed, 8 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\output.xml
Log: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests> |
```

## Test Utilisateur

[illegible]

## Commentaire

Les résultats obtenus nous montrent que tous les scénarios non passants engendrent des erreurs. Ce qui confirme l'hypothèse.

## Lab N°2 : Tests fonctionnels UI – Customer Service

À partir du document de référence “*Customer Service Test Cases*”, ci-joint les scripts automatisés et testés avec **Robot Framework**.

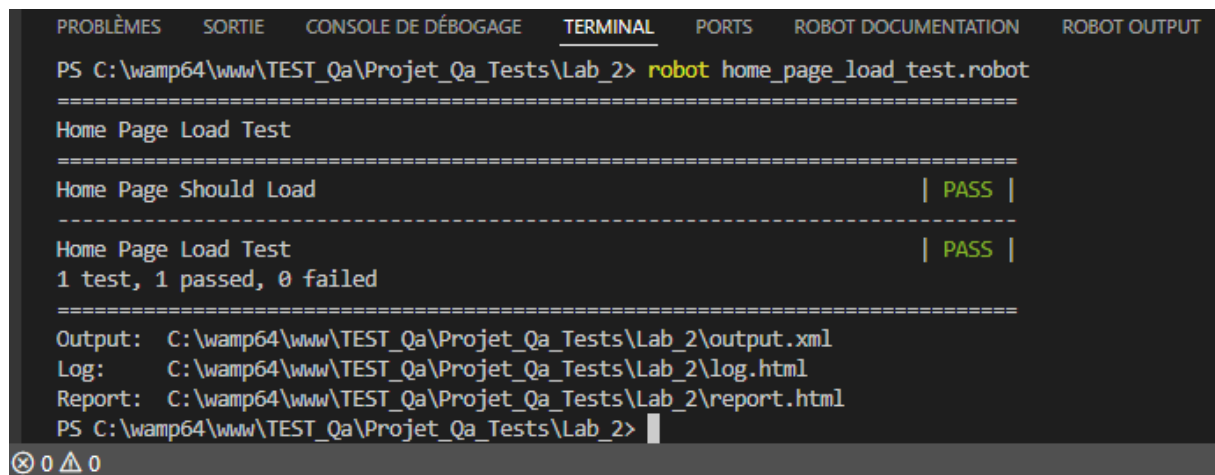
### Environnement de développement

Bien avant de pouvoir se lancer, dans la série des tests, rassurons-nous que Robot Framework est bien installé et à jour. Afin d'éviter que le navigateur que nous utilisons puisse planter et bloquer les tests, nous utilisons un outil intermédiaire qui va faire le lien entre le Framework et le navigateur.

Dans notre cas, l'outil est nommé : **GeckoDriver** et c'est lui qui se charge de faire le lien entre le **Robot Framework** et notre navigateur qui est **Firefox**, ici.

### CAS N°1 : Home page should load

Ici, il est question d'ouvrir le navigateur, d'accéder à la page d'accueil, de vérifier les contenus et de fermer le navigateur. Comme élément testé, on vérifie le titre de la page d'accueil. Tout est okay, d'après le résultat du script robot. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.



```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL  PORTS  ROBOT DOCUMENTATION  ROBOT OUTPUT

PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot home_page_load_test.robot
=====
Home Page Load Test
=====
Home Page Should Load                                     | PASS |
-----
Home Page Load Test                                     | PASS |
1 test, 1 passed, 0 failed
=====
Output:  C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:     C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report:  C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> |
```

### CAS N°2 : Login should succeed with valid credentials

Là, il est question d'ouvrir le navigateur, d'aller sur la page d'accueil, de cliquer sur **Login**, d'entrer des identifiants valides, de cliquer sur le bouton **Submit** et de fermer le navigateur. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL  PORTS  ROBOT DOCUMENTATION  ROBOT OUTPUT

PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot Login_should_succeed_with_valid_credentials.robot
=====
Login should succeed with valid credentials
=====
Login Should Succeed With Valid Credentials                                     | PASS |
-----
Login should succeed with valid credentials                                     | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> |
```

### CAS N°3 : Login should fail with missing credentials

Ici, il est question d'aller à la page de login, ne rien saisir, cliquer sur **Submit** et vérifier qu'il n'est pas connecté. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot Login_should_fail_with_missing_credentials.robot
=====
Login should fail with missing credentials
=====
Login Should Fail With Missing Credentials                                     .....== TITLE: Customer Service - Login
Login Should Fail With Missing Credentials                                     | PASS |
-----
Login should fail with missing credentials                                     | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> |
```

### CAS N°4 : "Remember me" checkbox should persist email address

Ici, il est question d'aller à la page de login, ne rien saisir, cliquer sur **Submit** et vérifier qu'il n'est pas connecté. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

**Attention :** Dans ce cas, la fonctionnalité « **Remember me** » n'a pas été implémentée dans l'application sur laquelle nous effectuons les tests. D'où le résultat obtenu ! Chose qui est tout à fait normal. Plus détails dans le code du script.

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot checkbox_should_persist_email_address.robot
=====
Checkbox Should Persist Email Address
=====
Remember Me Checkbox Should Persist Email Address                             | FAIL |
!= admin@robotframeworktutorial.com
-----
Checkbox Should Persist Email Address                                         | FAIL |
1 test, 0 passed, 1 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
```

### CAS N°5: Should be able to log out

Ici, il s'agit de vérifier qu'un utilisateur peut se connecter puis se déconnecter et qu'il arrive bien sur la page de déconnexion. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot Should_be_able_to_log_out.robot
=====
Should be able to log out
=====
Should Be Able To Log Out                                     | PASS |
-----
Should be able to log out                                     | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> █
```

### CAS N°6: Customers page should display multiple customers

Ici, il s'agit de vérifier que plusieurs clients sont visibles, dans la grille des contacts, après connexion. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot Customers_page_should_display_multiple_customers.robot
=====
Customers page should display multiple customers
=====
Customers Page Should Display Multiple Customers             | PASS |
-----
Customers page should display multiple customers             | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> █
```

### CAS N°7: Should be able to add new customer

Ici, il s'agit de vérifier que plusieurs clients sont visibles, dans la grille des contacts, après connexion. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot Should_be_able_to_add_new_customer.robot
=====
Should be able to add new customer
=====
Should Be Able To Add New Customer :: Connecte-toi et ajoute un no... | PASS |
-----
Should be able to add new customer                                     | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log:    C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> █
```



### CAS N°8: Should be able to cancel adding new customer

Ici, il s'agit de vérifier que plusieurs clients sont visibles, dans la grille des contacts, après connexion. Retrouvez les étapes, dans le code du script fourni, dans le dossier du projet. Pour chaque test, une capture d'écran est générée.

```
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> robot Should_be_able_to_cancel_adding_new_customer.robot
=====
Should be able to cancel adding new customer
=====
Should Be Able To Cancel Adding New Customer | PASS |
-----
Should be able to cancel adding new customer | PASS |
1 test, 1 passed, 0 failed
=====
Output: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\output.xml
Log: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\log.html
Report: C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2\report.html
PS C:\wamp64\www\TEST_Qa\Projet_Qa_Tests\Lab_2> |
```

Dans l'ensemble, les tests se sont bien déroulés !

## Lab 3 : Tests d'APIs REST – Intégration avec eBay

Voici quelques scénarios passant et non passant proposés, pour tester l'intégration avec l'**API Fulfillment d'eBay**.

Au préalable, il faut avoir :

- Un **compte développeur eBay** ;
- Un **Jeton OAuth valide** (scope requis : [https://api.ebay.com/oauth/api\\_scope/sell.fulfillment](https://api.ebay.com/oauth/api_scope/sell.fulfillment));
- Un **orderId valide** (pour les tests sur une commande existante) ;
- Une configuration API prête à envoyer des requêtes HTTP (ex. Postman, Robot Framework, Python Requests...).

### CAS N°1: createShippingFulfillment

#### Scénario Passant:

**Objectif** : Créer un fulfillment pour une commande valide.

**Méthode** : POST /order/{orderId}/shipping\_fulfillment

#### Attendu :

- Code HTTP 201 Created
- Contenu de la réponse avec un fulfillmentId valide.



**Scénario Non Passant:**

**Cas :** Données incomplètes ou invalides (ex : mauvais trackingNumber).

**Attendu :**

- Code HTTP 400 Bad Request
- Message d'erreur expliquant la ou les erreurs.

**CAS N°2: getShippingFulfillment****Scénario Passant:**

**Objectif :** Récupérer un fulfilment existant avec ID valide.

**Méthode :** GET /order/{orderId}/shipping\_fulfillment/{fulfillmentId}

**Attendu :**

- Code HTTP 200 OK
- Détails du fulfilment dans la réponse.

**Scénario Non Passant:**

**Cas :** fulfillmentId inexistant ou orderId erroné.

**Attendu :**

- Code HTTP 404 Not Found ou 403 Forbidden
- Message d'erreur : "Fulfillment not found" ou "Unauthorized access".

**CAS N°3: getShippingFulfillments****Scénario Passant:**

**Objectif :** Lister tous les fulfilments d'une commande.

**Méthode :** GET /order/{orderId}/shipping\_fulfillment

**Attendu :**

- Code HTTP 200 OK
- Tableau de tous les fulfilments liés à la commande.

**Scénario Non Passant:**

**Cas :** orderId inexistant ou non autorisé.

**Attendu :**

- Code HTTP 404 Not Found
- Message : "Order not found" ou "Invalid order ID".