**Conception des Tests**

**Projet :** Tester le site web de l’it-akademy «<https://inscription.it-akademy.fr/>»

**Objectif :** It-akademy souhaite valider les fonctionnalités présentent sur sont site web «https://inscription.it-akademy.fr/ » avant une mise en production :

* L’utilisateur pour s’inscrire à l'une des formations doit se connecter à l’écran de connexion « Plateforme admissions ». Si l’ utilisateur ne dispose pas de compte , il doit pouvoir créer son compte via l’écran « Nouveau compte ». Si l’ utilisateur a oublié son mot de passe , il peut demander une réinitialisation via l’écran « Mot de passe oublié ». une case à cocher permet aux utilisateurs d’indiquer au site de « se souvenir de moi ».
* le site Web doit s'afficher correctement dans différents navigateurs (IE, Firefox, Chrome, Safari et Opera). Les messages d’erreur ne doivent pas afficher des informations importantes.
* L'utilisateur une fois déconnecté du système ou si la session utilisateur a expiré, l'utilisateur ne devrait pas pouvoir naviguer sur le site.

**Environnement :**

* **Navigateurs** : IE, Firefox, Chrome, Safari et Opera
* **Systèmes d'exploitation** : Windows 11

**Règles de nommage :**

****

* Types de test :
* F Fonctionnel
* NF : Non Fonctionnel
* Fonctionnalité :
* CNC : Création Nouveau Compte
* CNX : Connexion au compte
* RMP : Réinitialisation de Mot de Passe
* Numéro du cas de test :
* T : Test1,2,3,4….etc.
* **Exemple :**



**Type de Tests :**

* Tests fonctionnels
* Tests de compatibilité
* Tests de sécurité

**Scénarios de tests :**

* Scénarios de tests fonctionnels:
* [Création de compte](https://docs.google.com/spreadsheets/u/0/d/18Cu9XFxY685egjIaGXeGQ0I546w4u3Cocdu5QOEvwIM/edit)
* [Connexion au compte](https://docs.google.com/spreadsheets/u/0/d/1Vxt9IVvHxp4EKc5q7Ob6PMyzpae0pOoit9VfKhO7uow/edit)
* [Réinitialisation de mot de passe](https://docs.google.com/spreadsheets/u/0/d/1IPkr9dixGBtwhKLrpEZ-HFjG9WRDqRoql-ZruUJyTPA/edit)
* Scénarios de tests de compatibilité (Non Fonctionnels) :
* [Création de compte NF](https://docs.google.com/spreadsheets/u/0/d/1uqoXfukdl0mbxWZ1Lga-swXtjuvSTjtqLoQoJFUA_ns/edit)
* [Connexion au compte NF](https://docs.google.com/spreadsheets/u/0/d/1zv6gcC0phc-hrobNc8RyMHTQhS45Wvgpc9X5qOqPNUE/edit)
* [Réinitialisation de mot de passe NF](https://docs.google.com/spreadsheets/u/0/d/1WcSW0bP3YHVpi1Ub5jTmthTNuoMhlD3RFjufqDLAtw0/edit)
* Scénario de test pour les tests de sécurité (Non Fonctionnels) :
* [Déconnexion de compte NF](https://docs.google.com/spreadsheets/u/0/d/1HLYtzJlihemIyXFx2K_F1rGYWSHCT5vaeQbMzLy4Q_0/edit)

**Synthèse de tests :**

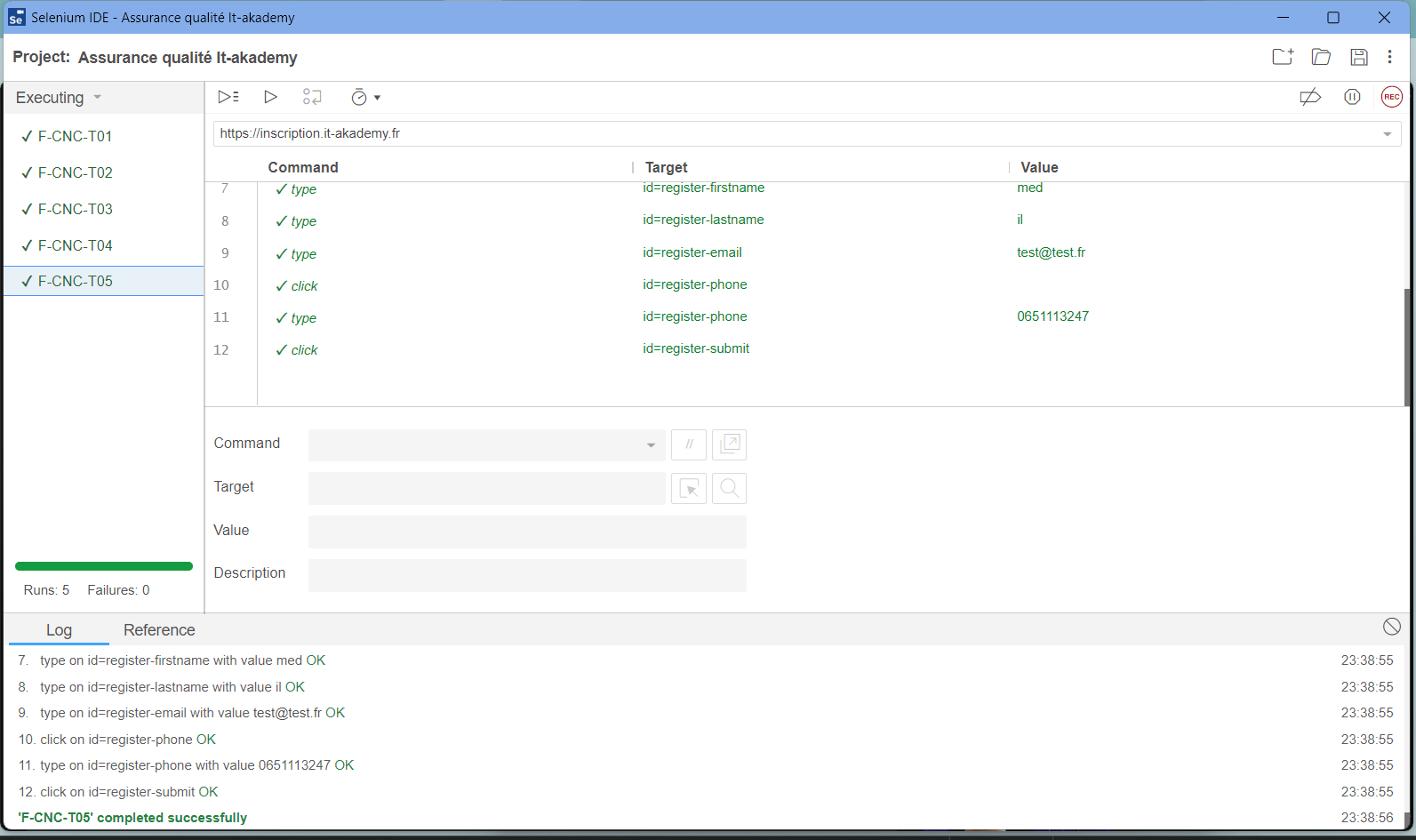
* [Tableau de tests](https://docs.google.com/spreadsheets/u/0/d/1JomfdsQ2ngJ3AQptDUMOjKgz6d4bzZO89oOr_0Feay0/edit)

**Date de début :** 23 Août 2024 **Date de fin prévue :** 02 Septembre 2024

**Automatisation des tests**

**Automatisation :**

* Automatisation du scénario de test [Création de compte](https://docs.google.com/spreadsheets/u/0/d/18Cu9XFxY685egjIaGXeGQ0I546w4u3Cocdu5QOEvwIM/edit) sur sélénium IDE :
* [Script du fichier JSON qui contient les informations sur les tests automatisés](https://docs.google.com/document/d/11Tqs-Ovd7YNdNnCoj_JShajJvnH6CfQ_squAvAjFb18/edit)
* Capture d’écran de l'exécution de l’automatisation sur Sélénium IDE :



* Exportation du script en *Python*:
* [Script généré par sélénium IDE](https://docs.google.com/document/d/1o3pO2ECqeuE0U7YmR-fVparJWEep2g5uVxVQJQoMvDw/edit)

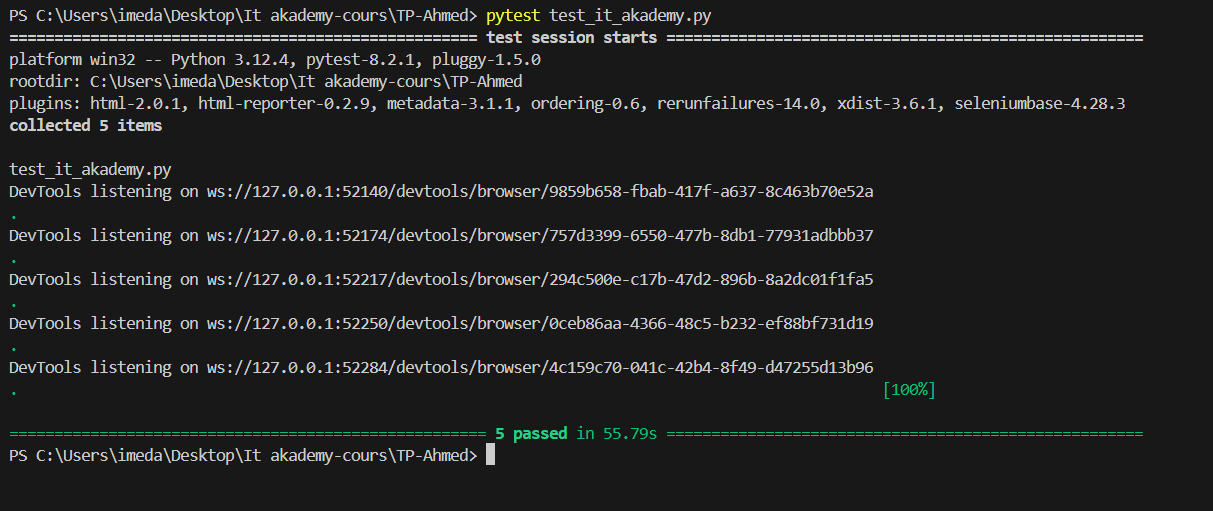
**Modifications :**

* J’ai ensuite apporté des modifications au script *Python* généré par sélénium IDE pour améliorer la robustesse des tests:
* [Script amélioré pour la robustesse des tests](https://docs.google.com/document/d/1L97yqqHRVkyk7syJcf0bLWRYBBecjbizVjtVbtO1aU4/edit)
* **les modifications apportées:**

1. **Ajout de Paramètres pour le navigateur :** Utilisation de chrome\_options avec l'argument --disable-search-engine-choice-screen pour désactiver l'écran de choix du moteur de recherche.
2. **Méthode de Fermeture des Cookies :** Ajout d'une méthode close\_cookies pour gérer la fermeture automatique des bannières de cookies, évitant ainsi des interruptions dans les tests.
3. **Mise en Pause :** Introduction de la méthode pause pour insérer des délais (pauses) dans le script, permettant de gérer les temps de chargement des pages.
4. **Utilisation de set\_window\_rect :** Remplacement de set\_window\_size par set\_window\_rect, permettant de définir plus précisément la taille de la fenêtre.
5. **Attente explicite avec WebDriverWait :** Introduction de l'attente explicite (WebDriverWait et EC.element\_to\_be\_clickable) pour s'assurer que les éléments sont cliquables avant de les utiliser, améliorant ainsi la fiabilité des tests.
6. **Gestion des Exceptions :** Ajout de la gestion des exceptions avec NoSuchElementException et TimeoutException pour éviter que le test ne plante en cas d'élément introuvable.
7. **Vérifications d'Assertions :** Ajout d'assertions pour vérifier que les champs de formulaire contiennent les valeurs attendues, renforçant la vérification de l'exactitude des tests.
8. **Vérification du Contenu de la Page :** Ajout d'une vérification pour s'assurer que le texte attendu est présent sur la page après certaines actions, améliorant la validation des résultats des tests.
9. **Suppression des Actions Inutiles :** Retrait des actions redondantes d'ActionChains qui étaient présentes dans le premier script, simplifiant ainsi le code.

**Pytest :**

* J’ai ensuite exécuté le fichier du script avec *Pytest* pour tester le bon fonctionnement des tests :



* Rapport HTML généré par le plugin “pytest-html”:

