Chapitre 3
Show Robot Framework

1. Du concret pour se motiver

Pour maîtriser efficacement un outil ou une technologie, l'une des approches les plus productives consiste à d'abord obtenir une vue d'ensemble de son potentiel et de ses capacités avant d'explorer ses fonctionnalités en profondeur. Cette méthode est particulièrement bénéfique pour introduire un outil d'automatisation des tests, dont le caractère fascinant repose en grande partie sur la capacité à exécuter des tâches de manière autonome.

Depuis la révolution industrielle, l'automatisation s'est intégrée dans divers domaines comme le textile, l'agriculture, l'industrie et les transports, et elle fait désormais partie intégrante de la vie quotidienne. Elle se retrouve, par exemple, dans les appareils domestiques tels que les aspirateurs et lave-vaisselle automatisés, dans les transports avec les systèmes de métro sans conducteur, ou dans les services bancaires avec les transferts automatiques de fonds. Les bénéfices de cette automatisation sont vastes : elle a permis des gains significatifs de productivité, une réduction des coûts de production et une optimisation des processus. Elle joue également un rôle clé dans l'exécution de tâches inaccessibles pour un humain, comme l'analyse de données massives en temps réel.

L'automatisation libère le temps humain en le déchargeant de tâches répétitives et routinières, ce qui permet aux individus de se consacrer à des activités à forte valeur ajoutée, là où la créativité et l'innovation sont irremplaçables. Elle contribue aussi à la sécurité humaine en prenant en charge certaines tâches dangereuses, comme le déminage pour détecter des engins explosifs ou la manipulation de substances toxiques dans les industries chimique et nucléaire.

Il est important de distinguer l'automatisation de l'intelligence artificielle, bien que les deux concepts soient souvent associés. L'automatisation est fondée sur la réalisation de tâches selon des règles préétablies, tandis que l'intelligence artificielle vise à imiter les capacités cognitives humaines. Contrairement à l'automatisation, l'intelligence artificielle inclut une capacité d'apprentissage et d'adaptation, permettant aux systèmes de s'améliorer par l'expérience et de répondre à des situations nouvelles sans programmation spécifique.

Ce chapitre se veut une source d'inspiration et de motivation pour une exploration approfondie, en offrant une immersion rapide et engageante dans le monde de l'automatisation dès les premières pages.

2. Première exécution

- Pour commencer, téléchargez le dossier sources_rbf_livre, disponible sur le site d'ENI ou via ce lien GitHub :
  https://github.com/ysidki/sources_rbf_livre
- Une fois le projet téléchargé, exécutez `windows\scripts\start.bat` sous Windows, ou `linux_macOS/Scripts/start.sh` sous Linux ou macOS. Pendant l'exécution du script, évitez toute interaction avec la machine, que ce soit via le clavier ou la souris.

Si tout se passe bien, le navigateur Chrome s'ouvrira sur la page d'accueil, et une série d'actions s'exécutera automatiquement :

– refus des cookies Google ;
– saisie dans le champ de recherche ;
– défilement dans la liste des résultats de recherche.

▶En cas d'erreur, vérifiez les points suivants :

– Python est installé avec l'option **Add python.exe to PATH** activée.

La commande `python` est bien reconnue depuis l'invite de commande ou le terminal.

Dans le projet exécuté, le navigateur Chrome a été spécifié dans les configurations. Si Chrome n'est pas déjà installé, Robot Framework le téléchargera automatiquement. Ce processus est détaillé dans le chapitre Création des tests - section Les bibliothèques Robot Framework, lors de la présentation de cette bibliothèque. Expliquons maintenant ce qui s'est produit après l'exécution du script start.bat (ou start.sh).

Le script contient une liste de commandes. Pour les curieux ou ceux ayant déjà une expérience avec Python, il est possible de consulter son contenu, bien que cela ne soit pas nécessaire pour la suite. Les commandes exécutées sont les suivantes :

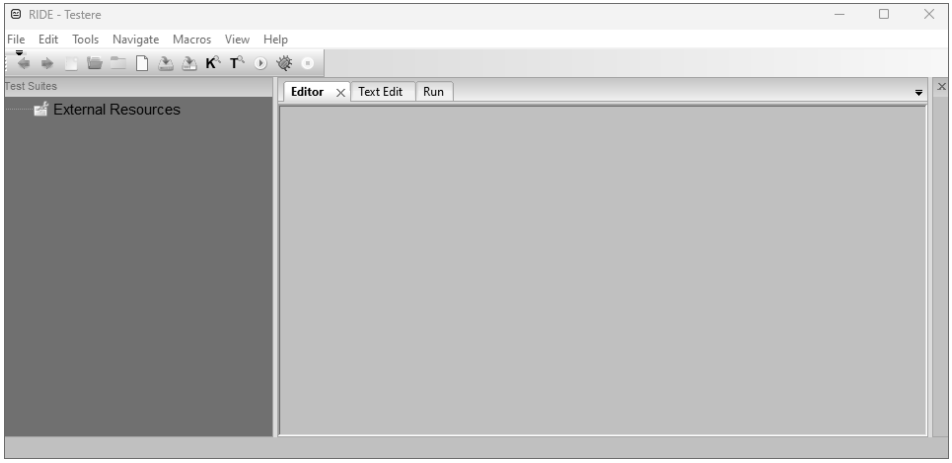
– création d'un environnement virtuel Python (suppression de l'ancien si nécessaire, puis recréation d'un nouveau) ;

– installation des bibliothèques Robot Framework requises pour configurer un environnement Robot Framework ;

– exécution d'une suite de tests Robot Framework nommée « test.robot », qui inclut un test simple simulant une recherche Google.

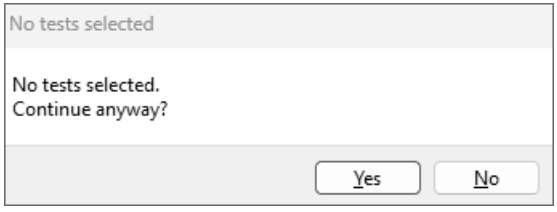
Ce livre guide dans la maîtrise de ces concepts, avec un focus particulier sur la création de suites de tests dans Robot Framework.

▶L'exécution peut également être réalisée depuis l'interface de l'éditeur RIDE, l'éditeur par défaut de Robot Framework. Pour lancer cette exécution via l'interface, exécutez le script `windows\start_ride.bat` sous Windows ou `linux_macOS/start_ride.sh` sous Linux ou macOS. Une fois le script exécuté, l'interface RIDE s'ouvre.



L'interface RIDE, l'éditeur par défaut de Robot Framework

- Pour continuer, allez dans **File** puis **Open Test Suite** et ouvrez le fichier test.robot. Ensuite, appuyez sur [F8] ou cliquez sur le bouton **Start** dans l'onglet **Run** pour lancer le test, qui exécutera une recherche sur Google dans le navigateur Chrome. Si un popup apparaît avant le début du test, validez-le simplement en cliquant sur **Yes**.



Popup affiché au moment de lancement des tests si aucun test n'est sélectionné

Dans certains environnements professionnels, des erreurs telles que **Fail to find chromedriver** peuvent survenir, souvent en raison de la configuration du proxy. De nombreuses entreprises implémentent un proxy pour des raisons de sécurité, ce qui peut bloquer certaines actions. Il est recommandé de vérifier auprès de l'administrateur réseau ou du service informatique que les connexions à localhost ne passent pas par un proxy et que le domaine google-chromelabs.github.io n'est pas bloqué.

Analyse de l'exécution : le premier test automatisé sous Robot Framework vient d'être réalisé. Ce test effectue une recherche simple sur Google et parcourt la liste des résultats. L'exécution a été volontairement ralentie pour observer les différentes actions effectuées, mais il est possible d'accélérer l'exécution des tests.

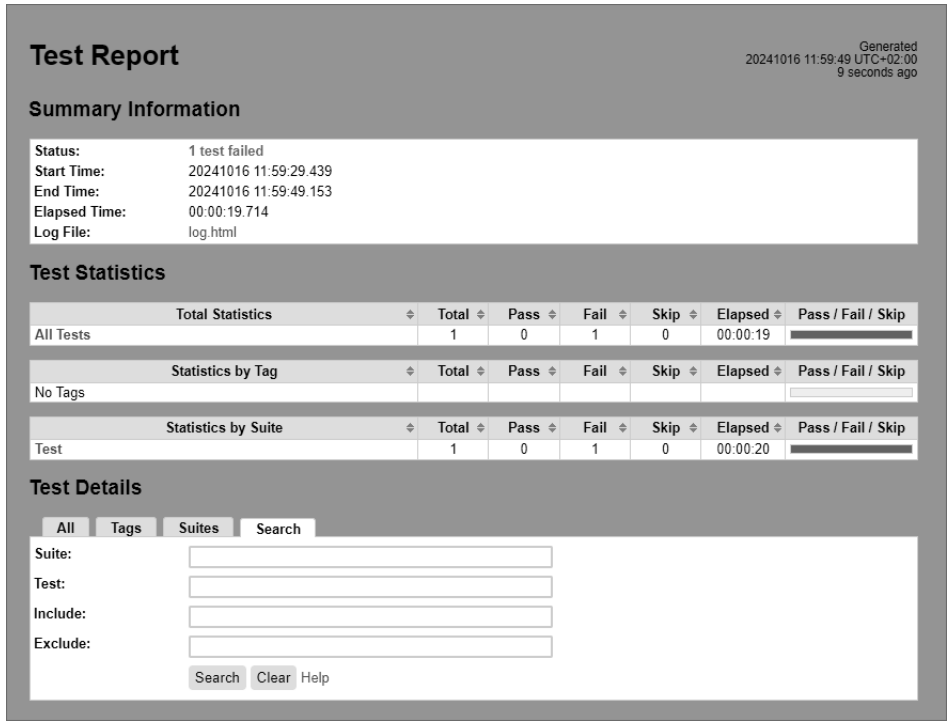
L'objectif de cette démonstration est de montrer que certaines actions, réalisées quotidiennement de manière manuelle, peuvent être automatisées. En réalité, cette simple présentation n'est que la partie émergée de l'iceberg. De nombreuses découvertes intéressantes et passionnantes se cachent encore derrière.

À la fin de l'exécution, certains fichiers sont générés automatiquement dans le dossier du projet. En ouvrant le fichier log.html par un double-clic, le rapport d'exécution s'affiche avec certaines étapes en vert, mais une en particulier en rouge. La couleur indiquée donne une idée du résultat des tests : en vert, cela signifie que l'étape a réussi, tandis qu'en rouge, cela indique que l'étape a échoué.

3. Première analyse d'un rapport

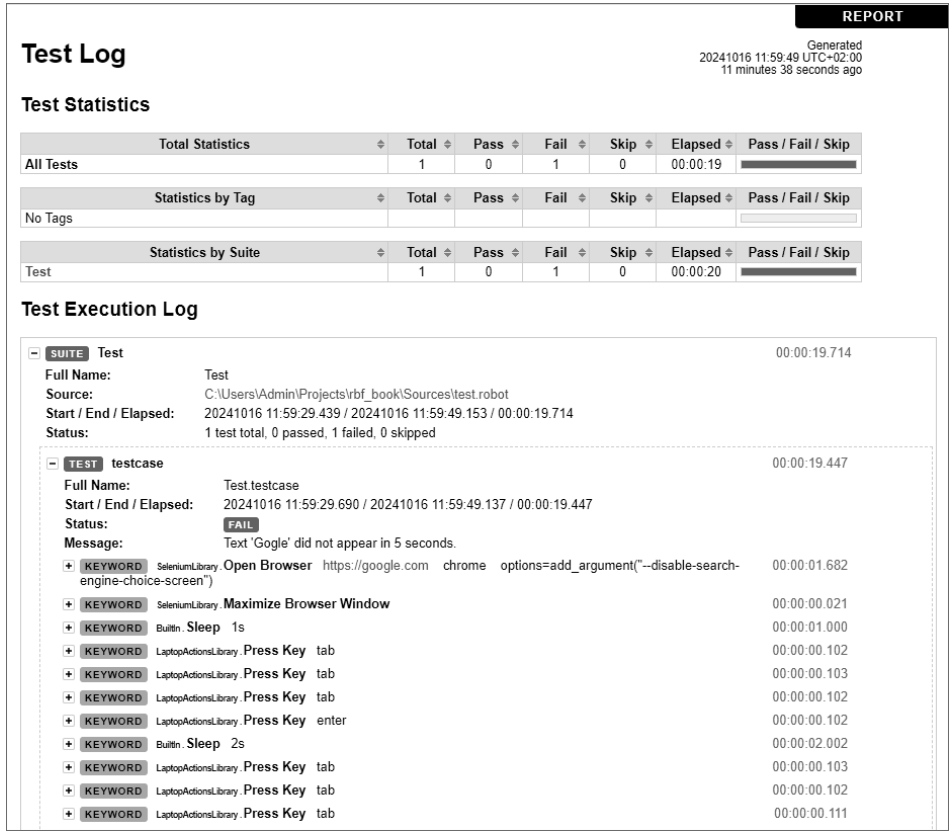
L'analyse de rapport impliquera principalement l'examen des fichiers report.html et log.html. Dans un premier temps, une attention particulière sera accordée au fichier report.html pour obtenir une vue d'ensemble des résultats de tests, avant de passer à l'examen des étapes de tests dans log.html pour identifier les étapes réussies et celles échouées. Une analyse approfondie de cette démarche sera détaillée dans la section Analyse des rapports du chapitre Exécution des tests.

Lors de la consultation du fichier report.html, une couleur rouge globale indique qu'au moins une étape de test a échoué. Si cette couleur rouge n'est pas visible, cela est probablement dû à l'utilisation d'un thème sombre ; passer au thème clair permet de la voir.



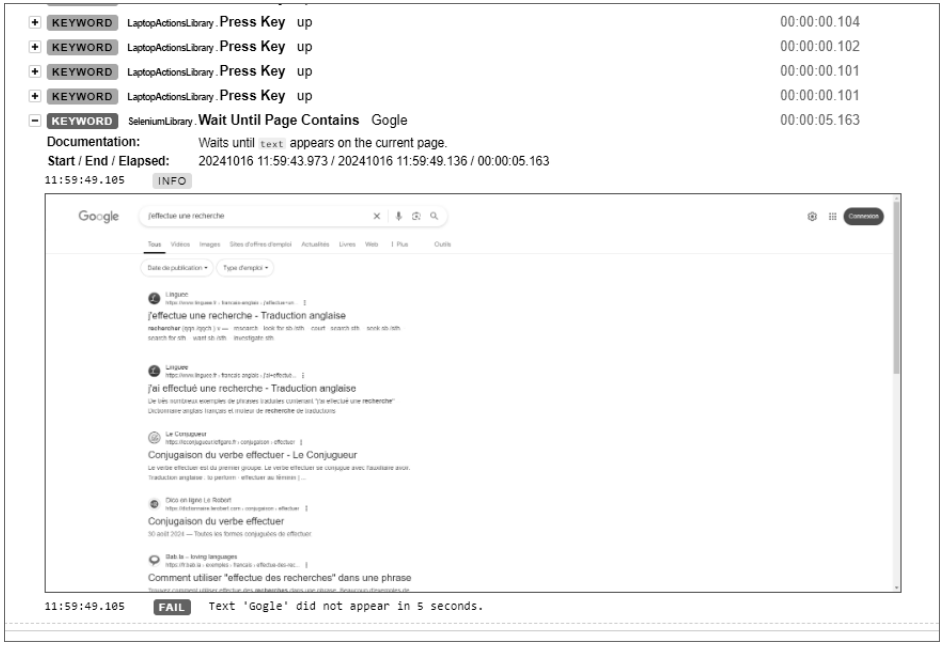
Vue globale du rapport Robot Framework

Pour examiner en détail les étapes de tests, le fichier log.html peut être ouvert. Une autre méthode, différente de l'ouverture directe de log.html, sera présentée dans la section Analyse des rapports du chapitre Exécution des tests.



Résultat de chaque étape de tests dans le fichier log.html

Une étape de test réussie s'affichera en vert, comme c'est le cas pour la première étape, qui consiste à ouvrir le navigateur. En revanche, une étape échouée apparaîtra en rouge. En défilant vers le bas de la page, il est possible de voir que la dernière étape, visant à vérifier la présence du texte « Gogle », a échoué. Lorsqu'une étape échoue, une capture d'écran accompagne celle-ci pour illustrer l'état de l'application au moment de l'échec.

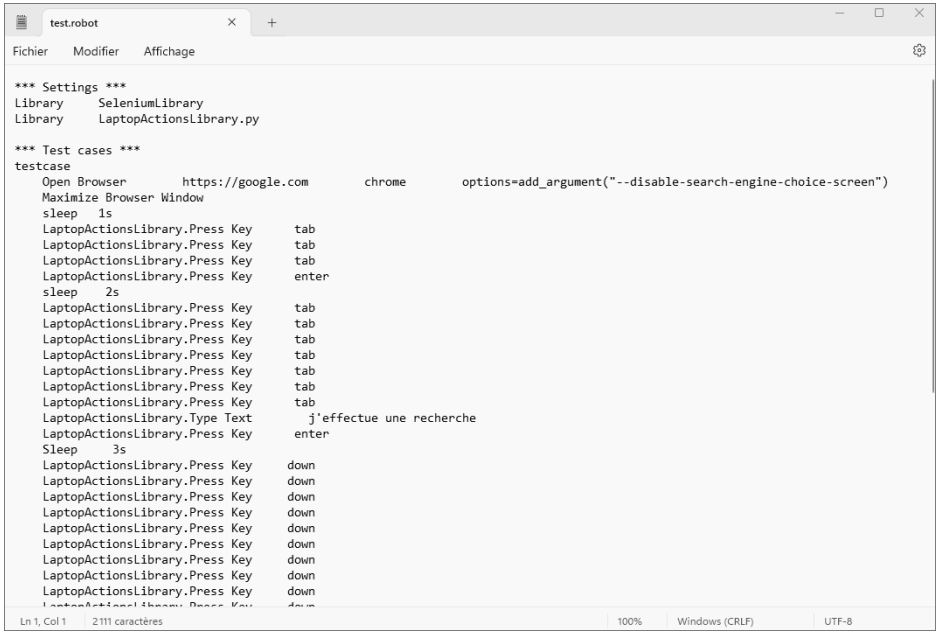


Échec de la dernière étape du test

Le message d'échec indique : **Text "Gogle" did not appear in 15 seconds**. En examinant attentivement ce message, la raison de l'échec du test devient évidente : une faute d'orthographe, où « Gogle » a été utilisé au lieu de « Google ». Cette étape constitue une analyse de rapport de test, une démarche typique effectuée par un testeur automaticien lors d'échecs de tests.

4. Première correction

Pour apporter une correction, l'accès au contenu du test est nécessaire. À ce stade, une méthode simple sera utilisée. Plutôt que d'accéder au test via RIDE, comme expliqué dans la section Éditeur RIDE du chapitre Exécution des tests, ou par un autre éditeur, comme abordé dans le chapitre Fonctionnalités avancées, section Éditeur RIDE vs les autres, cette section se contentera d'utiliser le Bloc-notes ou tout autre outil permettant d'ouvrir des fichiers texte. Les fichiers de tests Robot Framework, bien qu'ayant l'extension .robot, sont en réalité de simples fichiers texte. L'ouverture du fichier test.robot avec le Bloc-notes révèle le contenu suivant :



Contenu du fichier test.robot