|  |  |
| --- | --- |
| C:\Users\Romain\Downloads\selenium.jpg | SeleniumHQ  Browser Automatisation |

Tutoriel réalisé par :

Asma Moussavi

Romain Halter

# Selenium, l’élément chimique ?

On pourrait le penser mais nous allons parler de Selenium, un outil simple et puissant pour automatiser vos tests sur toutes vos applications web. Open source et gratuit, il est facile à comprendre et utiliser. De plus, il est écrit en Java, donc multiplateforme ! (Windows, Mac, Linux, Unix etc…). Selenium est capable de lire les pages HTML et d’interagir avec elles en toute autonomie. Enregistrer une séquence d’action (login, remplissage d’un formulaire …) et Selenium la rejouera pour vous autant de fois que vous le souhaitez en s’assurant du résultat (échec de login, champs incomplet …).

Selenium enregistre vos tests et dispose d’une fonction pour exporter ses tests dans plusieurs langages. Vous pourrez ainsi intégrer ceux-ci dans votre projet.

Les langages que Selenium supporte sont :

* Java
* C#
* Ruby
* Python
* PHP
* Pearl

Les navigateurs que supportent Selenium sont :

* Mozilla
* IE 7,8,9,10,11
* Google chrome
* Opera 8,9,10

Bien entendu, peu importe le langage utilisé pour réaliser votre application web, Selenium fonctionne principalement avec le code HTML et supporte le JavaScript (pop-up).

Selenium propose plusieurs outils :

* Selenium IDE : c'est une extension de Firefox, qui permet d'enregistrer une suite d'actions, qu'il sera possible de rejouer à volonté. . Il supporte les extensions régulières, les boucles, les états et de nombreuses autres fonctionnalités.
* Selenium WebDriver : il s'agit cette fois d'une API, disponible pour plusieurs langages, permettant de programmer des actions sur l'interface, et à vérifier les réponses. Les actions à réaliser peuvent être importées depuis Selenium IDE. Il prend également en charge Android et IOS.
* Selenium RC: c’est la version la plus ancienne de sélénium. RC travaille avec tous les langages de programmation mentionnés abve.
* Grille: Grille est utilisé pour exécuter des tests en parallèle sur plusieurs machines et navigateurs.

# Utilisation de Selenium

1. Mise en place du projet

Pour ce tutoriel, nous prendrons une application bancaire simple avec des comptes, des ajouts et retrait d’argent.

Installez le serveur **Tomcat** en le dézippant. (L’application, la base de données et les librairies sont déjà installées)

Lancer le **Tomcat** (« /bin/startup ») et rendez-vous sur l’url localhost:8080/jsp/test/bank

L’application contient une page d’authentification, d’enregistrement, de dépôt/retrait et de visualisation des opérations bancaires.

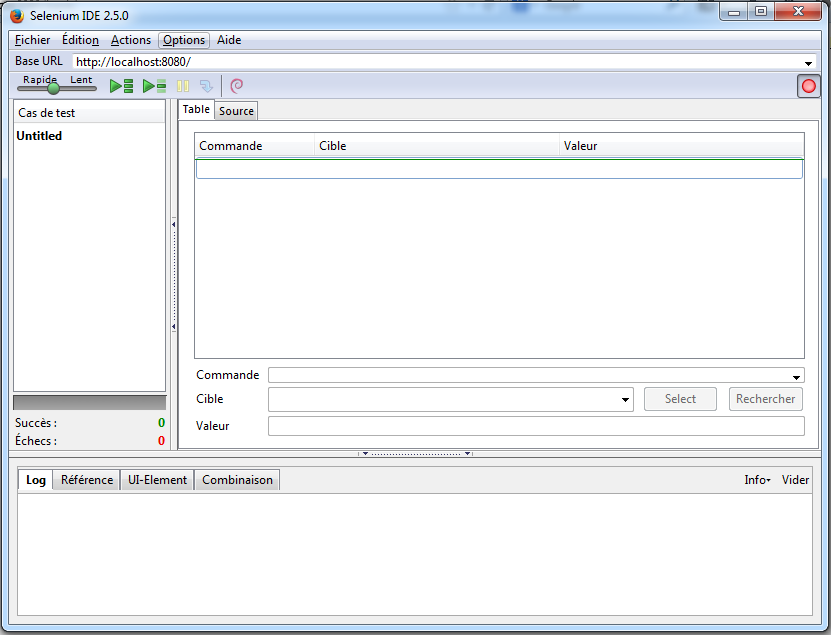
L’objectif des tests de l’application graphique est de s’assurer que le système d’authentification et les formulaires fonctionnent correctement. C’est une application bancaire, il ne faudrait pas qu’on puisse avoir accès à la page de quelqu’un d’autre …

1. Le plugin Firefox

Installez le plugin Selenium pour **Firefox :**

**http://release.seleniumhq.org/selenium-ide/2.5.0/selenium-ide-2.5.0.xpi**

Ce plugin ajoutera un bouton dans votre navigateur 

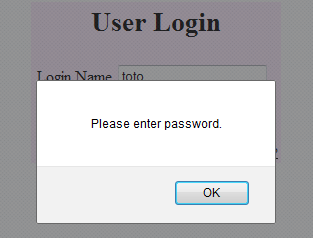
En cliquant dessus, vous ouvrez cette fenêtre :

Sur la page d’accueil du site, vous avez le choix entre vous identifier ou vous enregistrer. Nous allons créer un test afin de vérifier que l’authentification fonctionne.

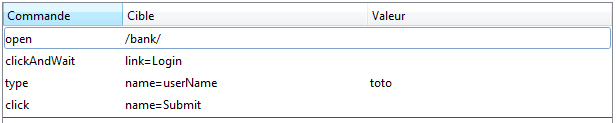
Selenium va enregistrer toutes votre action et les résultats afin de les rejouer automatiquement plus tard. Pour démarrer l’enregistrement, cliquer sur le bouton rouge  en haut à droite.

Cliquez sur login, entrez toto comme identifiant et appuyez sur le bouton login.

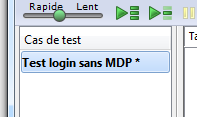
Vous devrez avoir ce message :



Recliquez sur le bouton d’enregistrement pour l’arrêter. Selenium a enregistré vos actions :



Votre test ne porte pas encore de nom, dans l’explorateur à gauche, clic droit sur votre test puis dans propriété, donnez-lui un nom significatif sur ce qu’il test. Exemple : Login sans mot de passe.

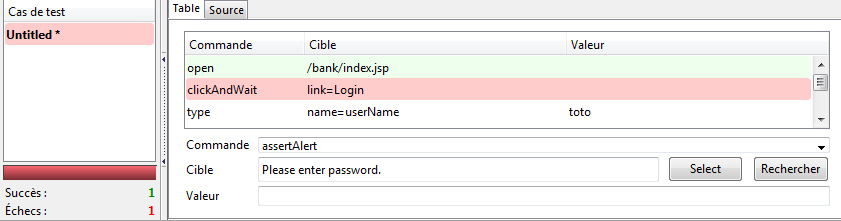


Vous pouvez désormais lancer le test grâce aux boutons verts juste au-dessus. Vous verrez votre navigateur bouger tout seul (en vitesse lente).

Connectez-vous avec l’identifiant **admin** et le mot de passe **admin**.

Relancez le test précédent.

Il se passe la chose suivante :



Le plugin Selenium vous indique un échec et arrête le test actuel.

Dans la fenêtre de log du plugin, vous pouvez voir le message suivant : 

Vous savez désormais que le lien « login » est manquant, en effet, une fois identifié, il n’apparait plus.

Nous allons maintenant vérifier qu’un texte est présent à un certain endroit.

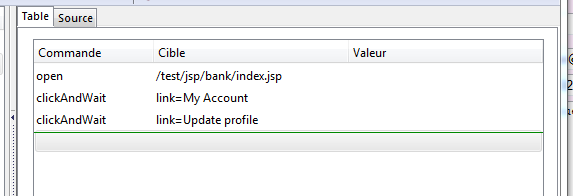
Créez un nouveau test et nommez le « test update profil »

Avec le compte admin, rendez-vous sur la page d’index. Commencez l’enregistrement et allez sur « my account » puis sélectionnez « update profil »

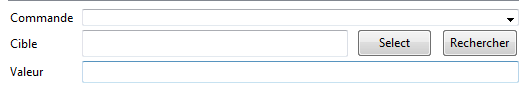
Les informations du compte devraient être auto-complétées.

Arrêtez l’enregistrement.

En dessous de la dernière commande enregistrée



Vous pouvez sélectionner la ligne vide sous le trait vert.

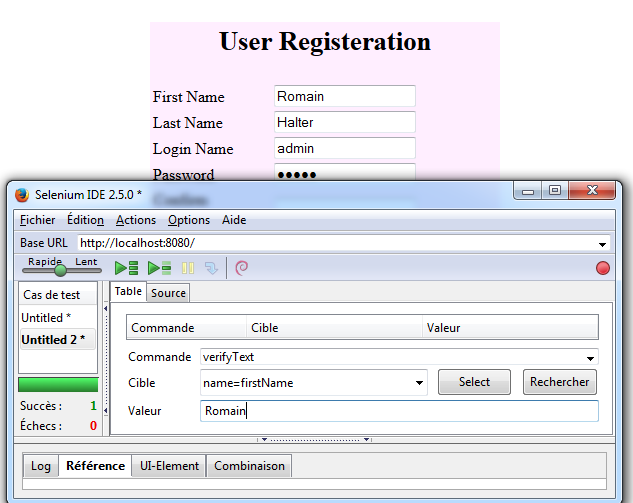


Dans le champ commande, tapez « verifyText » (L’auto-complétion fonctionne).

Ensuite pour sélectionner notre cible, cliquer sur le bouton « select »

Sélectionnez un champ, ensuite dans valeur, indiquez sa valeur.

Exemple :

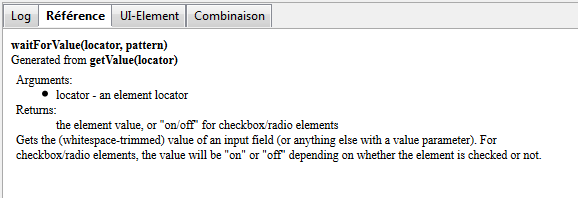


Vous pouvez également désigner un élément grâce à son id.

Lancez le test.

C’est un échec !

Et oui, il faut utiliser la commande « waitForValue » pour obtenir une valeur d’un élément input.



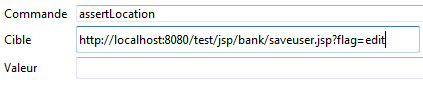
Vous trouverez une documentation dans l’onglet « référence » de la commande actuellement sélectionnée.

Remplacez « verifyText » par « «waitForValue ».

Relancez le test.

Nous allons désormais nous pencher sur l’URL. Comme vous avez pu le constater, la page « saveuser » sert à enregistrer et modifier un compte. Pour savoir le cas où nous sommes, un flag a été ajouté dans l’URL.

Dans le même test que précédemment, ajouter la commande et la cible suivante :

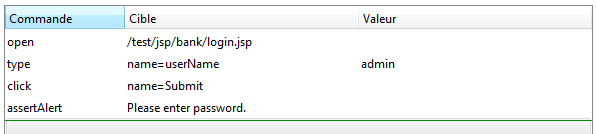


Elle permet de tester l’URL actuelle avec celle dans cible. Si elles sont différentes, le test sort en erreur.

Testez.

Maintenant, nous allons voir ce que Selenium peut faire avec du JavaScript.

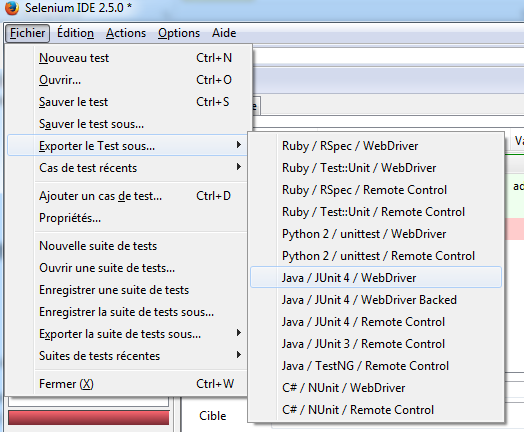
Réalisez le test suivant :



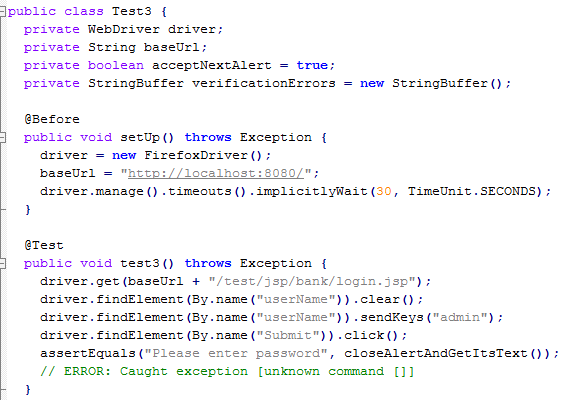
« assertAlert » permet de vérifier la présence d’une pop-up « alert » en JavaScript. Dans cible, mettez le message attendu.

1. Aller plus loin, insérer vos tests dans votre projet

Nous avons une suite de test. Mais ce n’est vraiment pas pratique d’ouvrir le navigateur et de charger les tests à chaque fois …

Selenium propose d’exporter vos tests :

Votre test est transformé en classe. Vous n’avez plus qu’à l’intégrer dans votre projet !



Merci d’avoir suivi ce tutoriel.

Selenium est puissant et possède encore bien d’autres fonctionnalités. N’hésitez pas à vous rendre sur <http://docs.seleniumhq.org/> pour plus d’informations.

**Bibliographie**

http://docs.seleniumhq.org/

<http://qtpselenium.com/selenium-tutorial/>

<http://www.jroller.com/selenium/#installation>

<http://www.objis.com/formation-java/tutoriel-selenium-installation-selenium-ide.html#partie2>

https://www.udemy.com/blog/selenium-ide-tutorial/