



WISEO

DIGITAL MAKERS

Tests fonctionnels automatisés :

Débuter avec Selenium IDE

16/03/2021

1. Brefs rappels sur les tests fonctionnels

2. Quels tests fonctionnels automatiser et pourquoi ?

3. Comment démarrer avec Selenium IDE ?

- 2.1 Résultats de test automatique

- 2.2 Actions principales et actions pratiques

- 2.3 Commandes

4. Exemples de scénarios écrits avec Selenium IDE

5. Démonstration

6. Recommandations



Brefs rappels sur les tests fonctionnels



Tests fonctionnels

RAPPELS SUR LES TESTS FONCTIONNELS (1/2)

λ Objectifs

- S'assurer que le comportement fonctionnel obtenu est conforme à celui attendu :
 - ✓ En déroulant un scénario composé d'étapes et
 - ✓ Pour chaque étape de vérification, en examinant la conformité attendu/obtenu d'une exigence.



➡ Pour valider l'application et obtenir une application de qualité.



➡ Pour instaurer ou entretenir un climat de confiance entre une société de services informatiques et ses clients.



λ Contexte

- Conditions réelles d'utilisation d'une application.
- Accès aux modifications 'techniques' (BDD, Code) inexistantes : mode utilisateur final.



λ 2 Types de tests fonctionnels

- Progression
 - ✓ Test réalisé sur une exigence qui vient être développée ou qui n'a pas encore été validée dans son ensemble.
- Non régression
 - ✓ Test réalisé sur une fonctionnalité déjà validée dans son ensemble,
 - ✓ Pour s'assurer que les évolutions ou corrections d'une nouvelle version de l'application n'ont pas entraîné de régressions sur cette fonctionnalité.



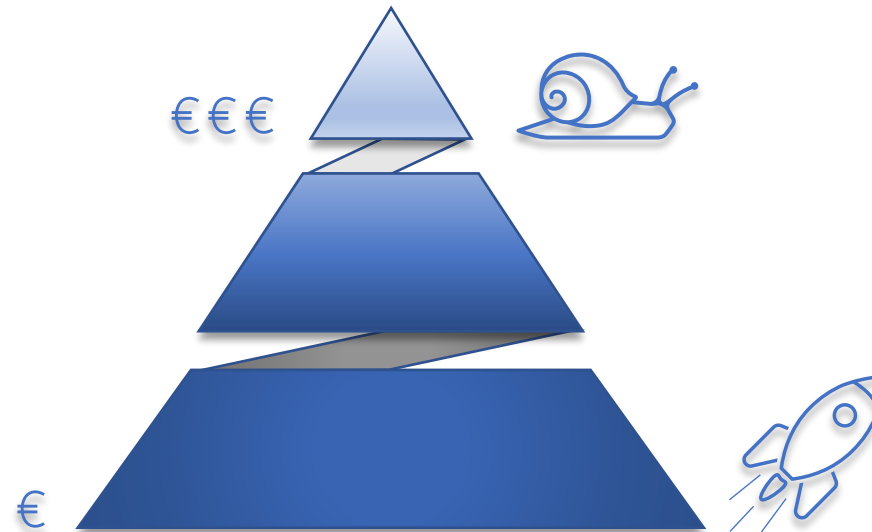
RAPPELS SUR LES TESTS FONCTIONNELS (2/2)

➤ Résultat d'un test fonctionnel

- Si au moins 1 étape échoue -> Tout le test est en erreur. -> La qualité n'est pas au rdv. ❌
- Si toutes les étapes passent sans erreur -> Le test est en succès. -> C'est un signe de qualité. ✅

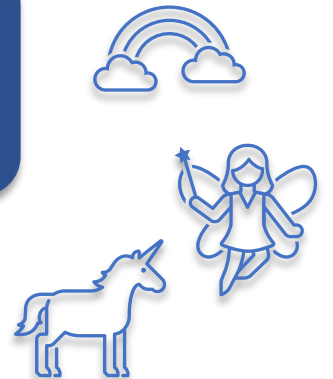
➤ Coût, rapidité et quantité des principaux tests

- Tests fonctionnels
- Tests d'intégration
- Test unitaires



- Plus coûteux
- Plus lents
- Moins nombreux
- Transverses
- Proches de la réalité

- Couteux
- Plus rapides
- Plus nombreux
- Peu réalistes



Quels tests automatiser et pourquoi



Tests fonctionnels

QUELS TESTS AUTOMATISER ET POURQUOI ? (1/3)

➤ Les critères d'acceptance des User Stories *(haut niveau)*

- Pourquoi ?
 - ✓ Ils portent la **valeur Métier** de l'application.
 - ✓ Il peut exister des régressions suite modification de code.
- Valeur ajoutée de l'automatisation : **Forte**

➤ les tests critiques

- Pourquoi ? l'automatisation des tests fonctionnels est **coûteuse**. Il faut automatiser les tests les plus importants.
- Rappel : la criticité d'un test est déterminée selon
 - ✓ Le **nombre** d'utilisateurs impactés
 - ✓ La **fréquence** d'utilisation de la fonctionnalité testée
 - ✓ La **valeur** Métier de la fonctionnalité testée
- Valeur ajoutée de l'automatisation : **Forte** si impossible unitairement
- Exemple / contre-exemple :
 - ✓ La **connexion** : touche tous les utilisateurs, selon leur fréquence d'utilisation. -> A automatiser ✓
 - ✓ Les tests de libellés n'ont aucune valeur Métier. -> *Ne pas automatiser* ✗

QUELS TESTS AUTOMATISER ET POURQUOI ? (2/3)

› les tests transverses

- Pourquoi ? Une action sur un écran impacte un ou plusieurs autres écrans : Test impossible unitairement.
- Valeur ajoutée de l'automatisation : **Forte**
- Exemple :
 - ✓ La création d'un catalogue entraînant la mise à jour d'une liste de catalogues.
 - ✓ L'ajout d'un participant à un événement entraînant la mise à jour des statistiques de participation

› les tests fastidieux

- Pourquoi ? Ils prennent du temps à réaliser à la main et/ou ils sont associés à des règles fonctionnelles complexes.
- Valeur ajoutée de l'automatisation : **Forte si l'automatisation est simple**
- Exemples :
 - ✓ Les **droits utilisateurs** : incontournables. Relativement simples mais longs à exécuter manuellement.
 - ✓ Un calcul de paie : nombreuses **règles complexes**

› les tests **CRUD** : Création, Consultation, Mise à jour, Suppression

- Pourquoi ?
 - ✓ C'est un standard de tests fonctionnels
 - ✓ Ils sont facilement automatisables
- Valeur ajoutée de l'automatisation : **Forte** (*même si partiellement réalisable unitairement côté Back et côté Front*)
- Exemples :
 - ✓ La création d'un catalogue d'un type précis entraînant l'exécution de nombreuses règles de gestion.
 - ✓ La suppression d'un catalogue, suite confirmation ou non.

QUELS TESTS AUTOMATISER ET POURQUOI ? (3/3)

λ les tests de recherche

- Pourquoi ?
 - ✓ C'est un standard des tests fonctionnels.
 - ✓ Facilement automatisables.
- Quand ? Après les tests CRUD
- Valeur ajoutée de l'automatisation : **Forte**
- Exemple : Recherche selon plusieurs critères

λ les tests d'erreur

- Pourquoi ?
 - ✓ C'est un standard des tests fonctionnels.
 - ✓ Souvent relégués au second plan si les cas passants sont OK. Mais essentiels pour les utilisateurs.
 - ✓ Facilement automatisables
- Quand ? Après les tests CRUD
- Valeur ajoutée de l'automatisation : **Moyenne +**
- Exemple : Une règle de gestion empêchant la création d'un catalogue. *(Vérification que l'objet n'est pas créé. Affichage d'un message d'erreur.)*

λ les tests d'annulation *(après confirmation ou non)*

- Pourquoi ?
 - ✓ C'est un standard des tests fonctionnels.
 - ✓ Qualifiés de « confort utilisateur » si les cas passants et d'erreur, sont OK. Mais essentiels pour les utilisateurs.
 - ✓ Facilement automatisables
- Rappel : L'utilisateur et à fortiori le client ont **droit aux erreurs de manipulation**.
- Quand ? Après les tests CRUD
- Valeur ajoutée de l'automatisation : **Moyenne**
- Exemple : L'annulation d'une création d'un catalogue.

QUELS TESTS NE PAS AUTOMATISER ET POURQUOI ?

λ les tests de tris

- Pourquoi ?
 - ✓ Ils sont compliqués à automatiser: ils doivent être réalisés manuellement.
 - ✓ Très peu de valeur Métier en général.
- Valeur ajoutée de l'automatisation :
 - ✓ **Minime**
- Exemple : Tri alphabétique des noms dans une liste.



λ les tests d'affichage

- Pourquoi ?
 - ✓ Ils sont compliqués à automatiser: ils doivent être réalisés manuellement.
 - ✓ Ils sont très subjectifs. (*Sauf s'ils sont flagrants*).
 - ✓ Très peu de valeur Métier en général.
- Valeur ajoutée de l'automatisation :
 - ✓ **Minime**
- Exemples : Texte tronqué, alignement de libellés.



λ les tests de traduction (*changement de locale*)

- Pourquoi ?
 - ✓ Une fois validée, la traduction est stable dans une application.
- Valeur ajoutée de l'automatisation :
 - ✓ Dépend du nb d'utilisateurs utilisant une langue différente de celle par défaut.






Présentation rapide de Selenium IDE



PRÉSENTATION RAPIDE DE SELENIUM IDE

› Selenium IDE

- Extension de plusieurs navigateurs Web.
 - ✓ A l'origine pour Firefox puis Chrome (Navigateurs officiels).
 - ✓ Disponible aussi pour Edge (Mais non officiel). 
- Permet d'enregistrer une suite d'actions sur une ou plusieurs pages Web. A rejouer à volonté.
- Utilisé pour créer des scripts simples.
- Exportables en différents langages :
 - ✓ Java/JavaScript
 - ✓ C#
 - ✓ Ruby
 - ✓ Python



SELENIUM IDE N'EST PAS SELENIUM



+

IDE

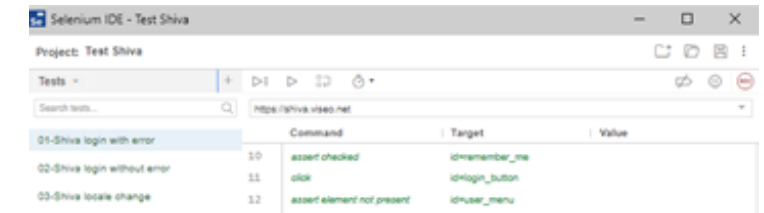
=



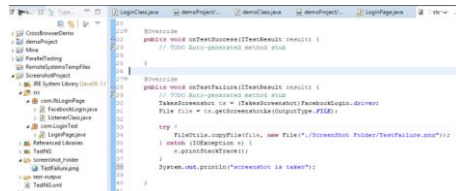
Selenium IDE

- ✓ **Selenium** est un framework de test informatique développé en Java.
- ✓ Il permet d'interagir avec différents navigateurs Web.
- ✓ Il facilite le test fonctionnel.

- ✓ Un **IDE** est un environnement de développement intégré (ou EDI en Français).
- ✓ Il assure l'automatisation de certaines tâches.



=> C'est une boîte à outils.
(Comme Eclipse)





Comment démarrer avec Selenium IDE ?



TÉLÉCHARGEMENT DES EXTENSIONS SELENIUM IDE

➤ Chrome

- Télécharger Selenium IDE depuis
 - ✓ <https://www.selenium.dev/downloads/> (Partie Selenium IDE)
 - ✓ [Selenium IDE - Chrome Web Store \(google.com\)](#)
 - <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=fr-fr>



➤ Firefox

- Extension Selenium IDE pour Mozilla Firefox
 - ✓ <https://www.selenium.dev/downloads/> ou
 - ✓ <https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/> ou
 - ✓ <https://addons.mozilla.org/fr/firefox/addon/selenium-ide/>



➤ Edge Chromium


- Télécharger Selenium IDE pour Edge (*Non officiel*)
 - ✓ <https://microsoftedge.microsoft.com/addons/detail/selenium-ide/ajdpfmkffanmkhejnopjppegokpogffp>

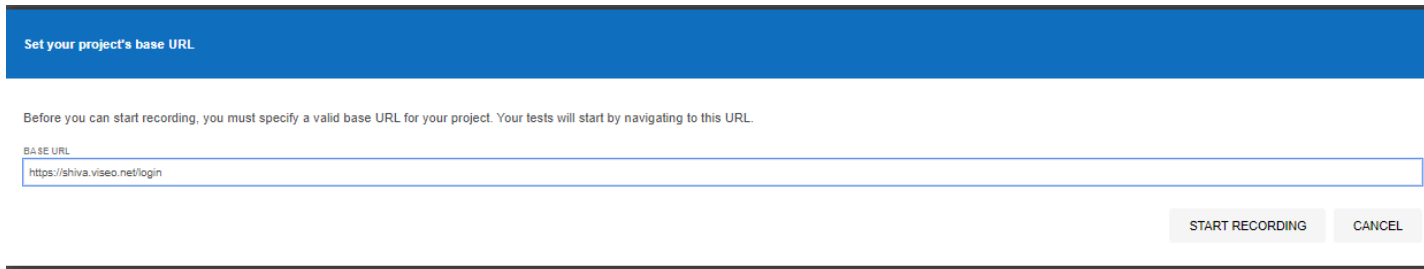
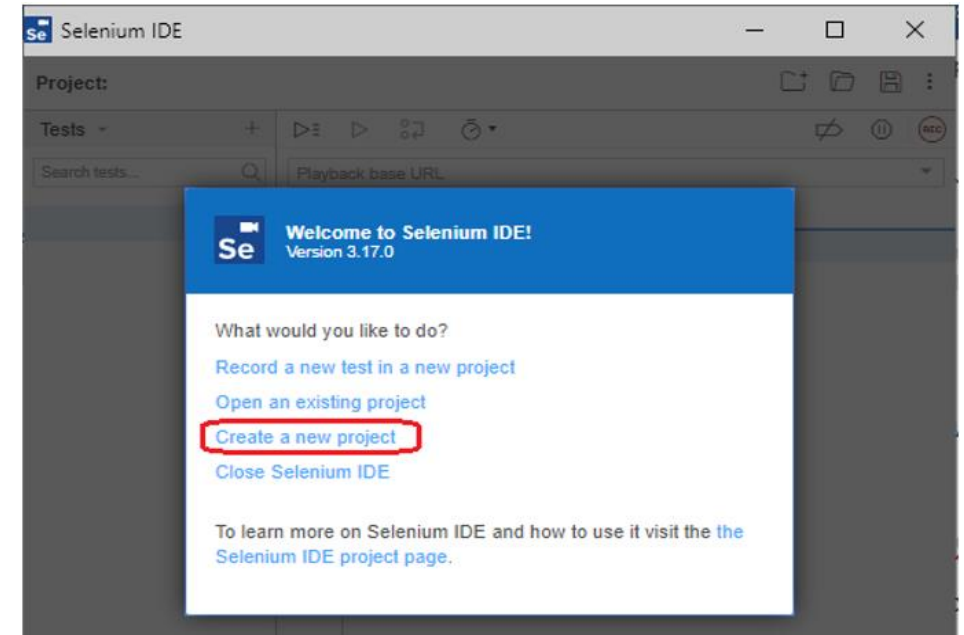


Puis lancez votre Navigateur Web. Une nouvelle icône apparaît :

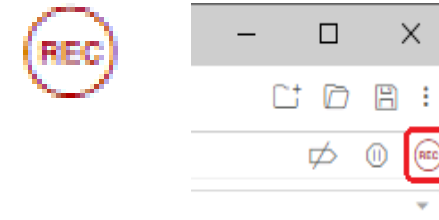


DÉMARRER SIMPLEMENT

1. Lancez votre navigateur. Positionnez-le sur la page Web à tester.
2. Cliquez sur l'icône Selenium IDE 
3. Sélénium IDE s'ouvre et demande 'What would you like to do' ? Choisissez 'Create a new project'.
4. Entrez le nom du projet de test automatisés.
5. Cliquez sur REC en haut à droite pour démarrer l'enregistrement du scénario de test, qui sera ensuite rejoué automatiquement
6. Précisez l'URL principale à tester, puis 'Start recording'.
7. Le navigateur Web s'ouvre sur la bonne page : vous pouvez commencer à enregistrer vos manipulations.

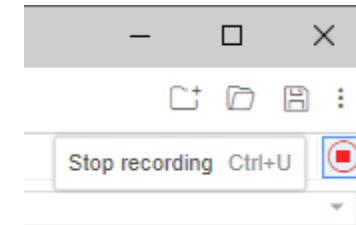


Exemple ici : *https://shiva.viseo.net/login*



DÉMARRER SIMPLEMENT

8. Pour arrêter l'enregistrement, il suffit de cliquer sur le carré rouge en haut à droite de Selenium IDE (derrière votre navigateur Web) et de renseigner un nom de scénario.




Name your new test

Please provide a name for your new test.

TEST NAME

Shiva login

You can change it at any time by clicking the  icon next to its name in the tests panel.

Et c'est tout!


➤ Consultez votre enregistrement

- Il est composé
 - ✓ d'une **liste** de commandes (click, type, etc)
 - ✓ sur une **cible** donnée (id d'un champ),
 - ✓ avec pour **valeur**, la valeur rentrée lors de l'enregistrement

Exemple : type id=username cvin123. Cette commande va entrer 'cvin123' comme nom d'utilisateur lors de la connexion.

Selenium IDE - Test Shiva

Project: Test Shiva

Tests 

Search tests...


https://shiva.viseo.net

	Command	Target	Value
1	open	/login	
2	set window size	1575x980	
3	click	id=username	
4	type	id=username	cvin123
5	click	id=password	
6	type	id=password	*****
7	click	id=login_button	

Command


type


//



Target

id=username





Value

cvin123

Description


LogReference

Test Shiva.side

Ouvrir un fichier

...

Afficher tout



DÉMARRER SIMPLEMENT

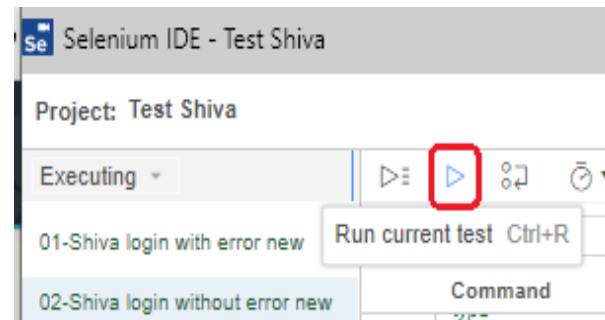
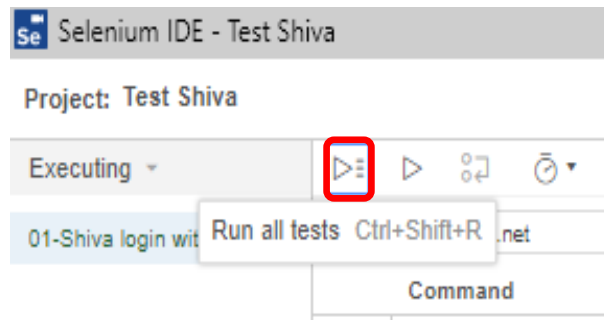
➤ Ajoutez des étapes de vérification

- Assert text ou verify text
- Assert ou verify element present / not present
- Etc

	Command	Target	Value
11	click	id=login_button	
12	assert element not present	id=user_menu	
13	click	css=.filter_error:nth-child(1)	
14	assert element present	css=.filter_error:nth-child(1)	
15	assert text	css=.filter_error:nth-child(1)	Le mot de passe est incorrect.
16	// ***** With wrong pwd, no access to Shiva *****		
17	assert element not present	css=#user_menu > li:nth-child(1)	
18	assert element not present	id=notification-list	
19	assert element not present	css=#test2 > div:nth-child(2)	//Aucune notification

➤ Vous pouvez rejouer le test autant que souhaité

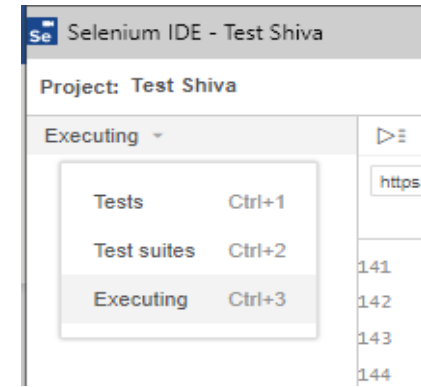
- Utilisez l'icône 'Run all tests' pour lancer tous les tests de la suite de tests.
- Utilisez l'icône 'Run current test' pour ne lancer que le test en cours de consultation.



RÉSULTAT DE TEST AUTOMATIQUE

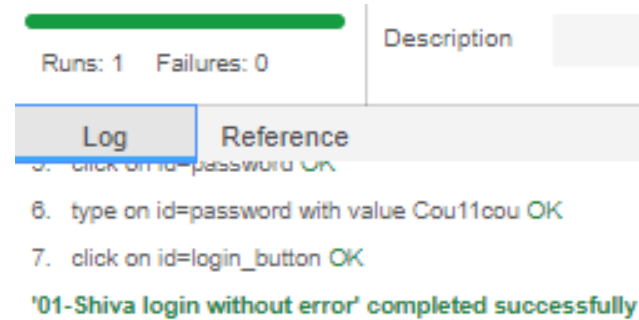
➤ nb de tests exécutés ensemble et le nb de tests en erreur

- sélectionnez le mode Executing, en haut à gauche de Selenium IDE
- Consultez « Runs : xx, Failures : yy », en bas à gauche.



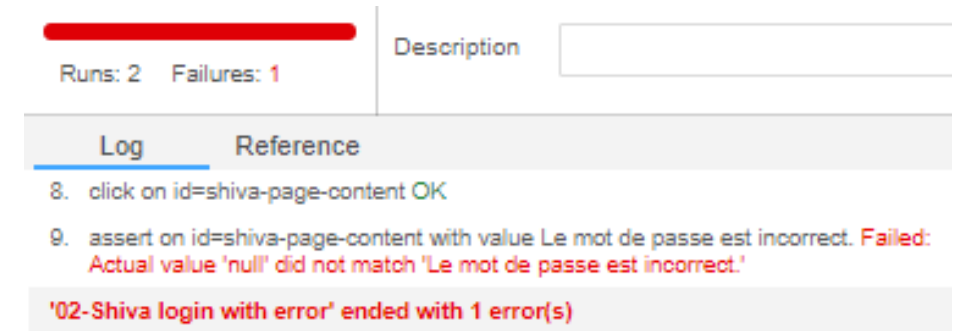
➤ Résultat OK

- Le scénario passe en vert
- Nb failures = 0
- Le log indique
 - ✓ le résultat du test, étape par étape : 'OK',
 - ✓ le résultat du test dans son ensemble.



➤ Résultat en erreur

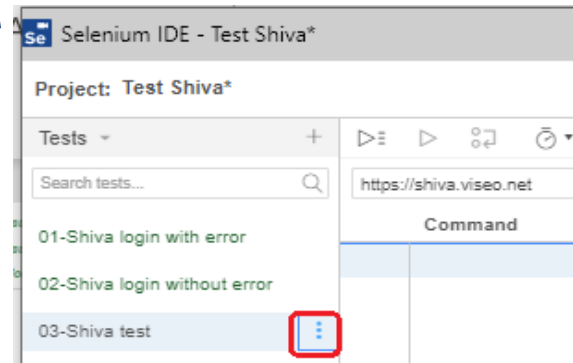
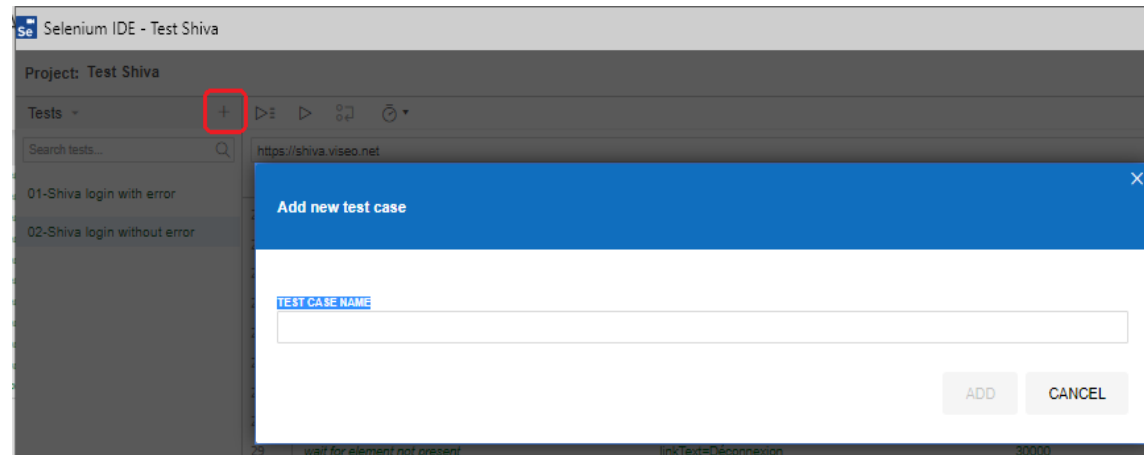
- Les étapes passées sans erreur passent en vert
- Les étapes en erreur passent en rouge
- Nb failures > 0
- Le log indique
 - ✓ le résultat du tests, étape par étape : 'Failed',
 - ✓ une raison de l'arrêt du test,
 - ✓ le résultat du test dans son ensemble.
- Interprétation : il existe un bug. Soit le test est mal écrit, soit le code correspondant dans l'application en cours de test, est en régression.



ACTIONS PRINCIPALES

Construction d'un test

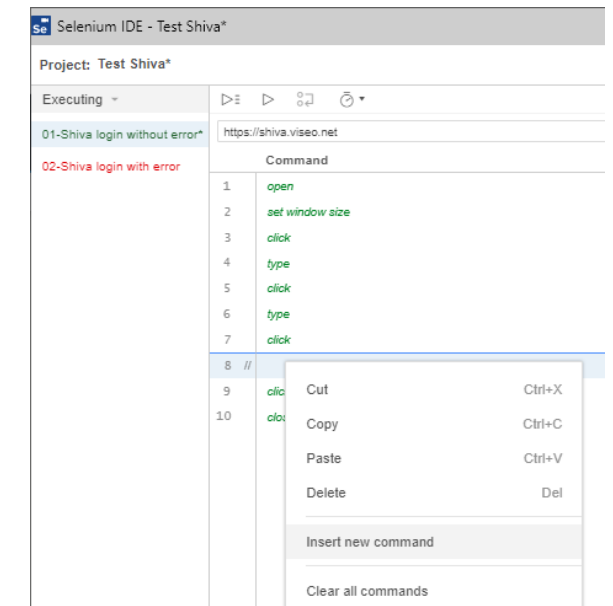
- Ajout d'un scénario
- Renommage
- Duplication
- Suppression
- Export
 - ✓ C# (Nunit, Xunit)
 - ✓ Java (JUnit)
 - ✓ JavaScript Mocha
 - ✓ Python pytest
 - ✓ Ruby RSpec



- Rename
- Duplicate
- Delete
- Export

Etapas de tests

- Ajout d'une commande
- Copie d'une commande
- Coupe
- Colle
- Suppression



ACTIONS PRATIQUES

› Rédaction d'un test

- Pour mettre en **commentaire** une ligne de commande :

Command	click	//	
Target	id=username		
Value			
Description			

› Exécution d'un test

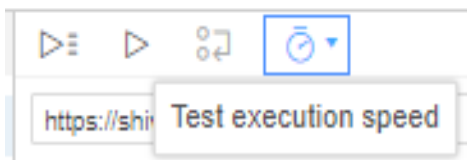
- Pour lancer une partie de test (clic droit sur une étape de test)



Toggle breakpoint	B
Play to this point	S
Record from here	U
Play from here	

- Evite de jouer tout le test
- Gain de temps pour valider la rédaction du test

- Pour adapter la **vitesse** d'exécution d'un test



- Rapide
 - Lorsque la construction du test est terminée
 - Gain de temps
- Lent
 - En cours de construction du test

QUELQUES COMMANDES SELENIUM IDE

Commande	Explication
Assert alert	Confirme qu'un message d'erreur s'affiche.
Assert checked / not checked	Confirme que l'élément cible a été coché (ou non).
Assert element present / not present	Confirme que l'élément cible est (ou n'est pas) présent dans la page.
Assert selected value / assert selected label	Confirme que la valeur sélectionnée dans une liste (dropdown) est de la valeur attendue (ou non).
Assert text	Confirme que l'élément cible contient le bon texte.
Assert title	Confirme que le titre de la page courante contient le bon texte.
Check / uncheck	Coche (décoche) une case à cocher ou un radiobouton.
Click	Clique sur l'élément cible (lien, bouton, case à cocher, radio bouton etc).
Close	Ferme la fenêtre courante.
Execute script	Exécute un script javascript (cible) et stocke le résultat dans une variable (value). Pour ensuite faire des boucles if/else ou while.
If / else / end	Si la condition est vérifiée, exécute la commande qui suit, sinon exécute la commande après le 'else', jusqu'à la commande 'end.' S'utilise avec une variable, résultat d'exécution d'un script (commande exécute script) ou stockée (commande store).
Open	Ouvre une page Web à partir de son URL (relative ou absolue) et attend que la page soit chargée avant de procéder à la suite du test.
Pause	Temporise le test entre deux actions. Target = temps d'attente en ms.
Select	Sélectionne un élément d'une liste (dropdown), grâce à un type de localisateur (Ex. label=, value=, id=, index=). S'il n'y a pas de type de localisateur, une correspondance sur le libellé sera essayée.
Store	Stocke une valeur (target) dans une variable (value).
Type	Entre une valeur dans un champ.
Verify checked / not checked	Vérifie (sans bloquer) qu'une case à cocher ou un radio-bouton est coché(e) ou non.
Verify element present / not present	Vérifie (sans bloquer) que l'élément cible est (ou n'est pas) présent sur la page Web.
Wait for element present / not present	Vérifie qu'après un laps de temps donné (value en ms), l'élément cible devient disponible (ou n'est plus disponible).

À retrouver sur <https://www.seleniumhq.org/selenium-ide/docs/en/api/commands/>



Exemples de scénarios



EXEMPLE : LA CONNEXION

➤ Connexion impossible

- Objectif du test : se connecter avec une login/mot de passe incorrects, ne permet pas d'accéder à l'application.

	Command	Target	Value
1	//	***** Try connexion with wrong passord : returns an error *****	
2	open	https://shiva.viseo.net/login	
3	set window size	945x1020	
4	click	id=username	
5	type	id=username	cvin123
6	click	id=password	
7	type	id=password	12345
8	click	id=remember_me	
9	check	id=remember_me	
10	assert checked	id=remember_me	
11	click	id=login_button	
12	assert element not present	id=user_menu	
13	click	css=.filter_error:nth-child(1)	
14	assert element present	css=.filter_error:nth-child(1)	
15	assert text	css=.filter_error:nth-child(1)	Le mot de passe est incorrect.
16	assert element not present	css=#user_menu > li:nth-child(1)	
17	assert element not present	id=notification-list	
18	assert element not present	linkText=Merci d'effectuer vos validations dès que possible	
19	assert element not present	css=#test2 > div:nth-child(2)	Aucune notification
20	assert element not present	id=ftm	
21	assert element not present	id=conge	
22	assert element not present	id=delegations	
23	assert element not present	id=ded_menu	
24	assert element not present	id=validation_resp_eotp	
25	assert element not present	id=projet	
26	//	close	

EXEMPLE : LA CONNEXION

➤ Connexion possible – Accès à l'application

- Objectif du test : se connecter avec une login/mot de passe corrects, permet d'accéder à l'application.

	Command	Target	Value
1	if ***** Connexion with correct password : opens Shiva *****		
2	open	https://shiva.viseo.net/login	
3	set window size	945x1020	
4	click	id=username	
5	type	id=username	cvn123
6	click	id=password	
7	*** type id=password *** Pwd OK ***		
8	click	id=login_button	
9	click	css=#user_menu > li:nth-child(1)	
10	click	css=strong	
11	click	id=header_wrapper	
12	click	id=locale	
13	select	id=locale	label=French
14	assert element not present	css=.filter_error:nth-child(1)	Le mot de passe est correct
15	assert element present	id=notification-list	
16	click	id=notification-list	
17	if store	false	isFinMois
18	execute script	return "false"	isFinMois
19	assert	isFinMois	false
20	if	\$(isFinMois) === "true"	
21	assert element present	linkText=Merçi d'effectuer vos validations dès que possible	
22	verify text	linkText=Merçi d'effectuer vos validations dès que possible	Merçi d'effectuer vos validations dès que possible
23	else		
24	assert element present	css=#test2 > div:nth-child(2)	
25	verify text	css=#test2 > div:nth-child(2)	Aucune notification
26	end		
27	assert element present	id=ftm	
28	assert element present	id=coonge	
29	assert element present	id=delegations	
30	assert element present	id=dad_menu	
31	assert element present	id=validation_resp_eotp	
32	assert element present	id=projet	
33	assert element present	linkText=Déconnexion	

EXEMPLE : TRADUCTION

➤ **Objectif du test** : la traduction de l'application est possible

	Command	Target	Value
1	// ***** Translations *****		
2	open	/	
3	set window size	945x1020	
4	select	id=locale	label=Français
5	assert selected label	id=locale	Français
6	click	id=user_menu	
7	click	css=#user_menu > li:nth-child(1)	
8	assert text	css=#user_menu > li:nth-child(1)	Connecté en tant que CORINNE VIN (cvi3089) - Déconnexion
9	click	css=#test2 > div:nth-child(2)	
10	click	css=strong	
11	assert text	css=strong	CORINNE VIN (cvi3089)
12	click	id=notification-list	
13	click	css=h3	
14	assert element present	id=ftm	
15	assert text	id=ftm	Activité
16	assert element present	id=conge	
17	assert text	id=conge	Congés
18	select	id=locale	label=Anglais
19	assert selected label	id=locale	English
20	click	id=user_menu	
21	click	css=#user_menu > li:nth-child(1)	
22	assert text	css=#user_menu > li:nth-child(1)	Logged in as CORINNE VIN (cvi3089) - Logout
23	assert text	id=ftm	Activity
24	assert text	id=conge	Vacations
25	select	id=locale	label=French
26	assert selected label	id=locale	Français
27	// click	linkText=Déconnexion	
28	// wait for element not present	linkText=Déconnexion	30000
29	// close		

EXEMPLE : IF ELSE END

➤ **Objectif du test :** passe dans l’une ou l’autre des boucles selon vérification ou non d’une condition

- Store

Command	Target	Value
store	false	isFinMois

- Ou Execute script

Command	Target	Value
execute script	Return “false”	isFinMois

- Puis If else end

Command	Target	Value
if	\$(isFinMois) === “true”	
assert element present	linkText=Merci d’effectuer vos validations dès que possible	
else		
assert element present	css=#test2 > div:nth-child(2)	
verify text	css=#test2 > div:nth-child(2)	Aucune notification
end		

EXEMPLE : CHANGEMENT DE FENÊTRE WINDOWS

➤ **Objectif du test :** ouvrir une fenêtre puis revenir sur la précédente

Command	Target	Value
click	css=#held > img	//clique sur le bouton d’aide
store window handle	root	
select window	handle=\${win2306}	//Ouvre la fenêtre d’aide à partir de la fenêtre principale de l’application
close		//Ferme la fenêtre d’aide
select window	handle=\${root}	//Retourne sur la fenêtre principale



Démonstration



DÉMONSTRATION

Le scénario : connexion avec un mauvais login/pwd

- Enregistrement automatique d'un test avec Selenium IDE

- Ajout de la commande assert text

	Command	Target	Value
1	open	/login	
2	set window size	945x1020	
3	click	id=username	
4	type	id=username	cvi123
5	type	id=password	123
6	click	id=login_button	
7	click	css=.filter_error:nth-child(1)	
8	assert text	css=.filter_error:nth-child(1)	Identification incorrecte.

Commandes enregistrées automatiquement sans les commandes **assert** ou **verify**

Commande de vérification ajoutée manuellement

Exécution d'un test

Erreur :



- Running '12-demo'16:01:45
- 1.open on /login OK16:01:46
- 2.setWindowSize on 945x1020 OK16:01:46
- 3.click on id=username OK16:01:46
- 4.type on id=username with value cvi123 OK16:01:46
- 5.type on id=password with value 123 OK16:01:47
- 6.click on id=login_button OK16:01:47
- 7.click on css=.filter_error:nth-child(1) OK16:01:47
- 8.assertText on css=.filter_error:nth-child(1) with value Identification incorrecte Failed:16:01:47
- Actual value "Identification incorrecte." did not match « Mauvaise identification."
- '12-demo' ended with 1 error(s) 16:01:47

Pas d'erreur :



- Running '12-demo'16:02:27
- 1.open on /login OK16:02:27
- 2.setWindowSize on 945x1020 OK16:02:27
- 3.click on id=username OK16:02:27
- 4.type on id=username with value cvi123 OK16:02:28
- 5.type on id=password with value 123 OK16:02:28
- 6.click on id=login_button OK16:02:28
- 7.click on css=.filter_error:nth-child(1) OK16:02:28
- 8.assertText on css=.filter_error:nth-child(1) with value Identification incorrecte. OK16:02:28
- '12-demo' completed successfully 16:02:29

DÉMONSTRATION

➤ Autres scénarios sur Shiva

- Connexion correcte
- Changement de locale (Traductions)
- FTM
- Congés
- Délégations
- Demandes d'engagement de dépenses
- Responsable EOTP
- Projets
- Transverse (Aide)



Certifié en quelques minutes : un robot ne se pose pas de questions.





Recommendations



RECOMMANDATIONS (1/3)

› Quand commencer l'automatisation ?

- Ne commencez pas l'automatisation de tests fonctionnels tant que l'application ne commence pas à se figer sur un ou plusieurs modules.
- Sinon, c'est trop tôt. Le coût de maintenance sera trop élevé.



› Tests avec valeur Métier et Tests critiques

- N'automatisez que les tests avec de la valeur et les tests les plus critiques
✓ Selon la *difficulté de réalisation manuelle* du test

› Ordre des tests

- Attention à l'ordre des tests. Pensez au bon enchainement des scénarios.
- Exemple : des scénarios dépendants.

› Données

- N'utilisez que des données existantes stables ou
- Pour lesquelles l'utilisateur est autonome en création/modification et suppression.

› Performance

- Attention à la performance de l'application.
- Exemple : évitez de vous déconnecter et de vous reconnecter entre chaque test.

Suite properties

Timeout (seconds)

Run in parallel ☐

Running in parallel works only in the runner

Persist session ☒

RECOMMANDATIONS (2/3)

➤ Commande open

- Utilisez la commande **open** avec une URL en entier, plutôt qu'une URL relative.
- Pour éviter d'ouvrir le navigateur, sur la bonne page, à chaque exécution de test.
- Exemple : open 'https://shiva.viseo.net/login' plutôt que open '/login'

➤ Bonne pratique de rédaction

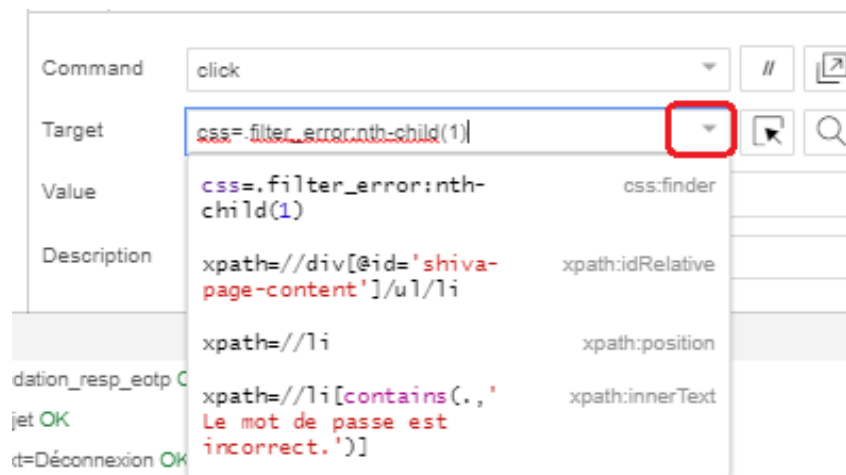
- Pour chaque étape de vérification (assert / verify), commencez par implémenter un test de façon à ce qu'il tombe en erreur. Puis implémentez le test correct (sans erreur).

➤ Assert vs verify

- Utilisez **assert** lorsqu'une étape de vérification est **bloquante**.
 - ✓ L'exécution du test **s'arrêtera net**.
 - ✓ Le test tombera en erreur.
- Utilisez **verify** lorsque l'étape est **non bloquante**. (Vérification de titre, de libellés, etc.).
 - ✓ L'exécution du test **se poursuivra**.
 - ✓ Le test tombera en erreur.

➤ Flèche de détail d'une cible

- Elle contient du texte sur une cible pour des commandes telles que **click** ou **type**.
- Utilisez la pour vous y retrouver après l'enregistrement d'un scénario.
- Puis ajoutez une commande **verify** ou **assert** sur cette cible.
- Exemple ci-contre :
 - ✓ Après la commande '**click** css=filter_error:nth(child(1))', enregistrée par Selenium IDE,
 - ✓ ajoutez une commande '**assert text** css=filter_error:nth(child(1)) **Le mot de passe est incorrect.**'.



RECOMMANDATIONS (3/3)

➤ Cibles css

- Evitez d'utiliser les cibles commençant par **xpath**. Préférez-leur des cibles commençant par **css**, plus générales en cas de modification du code applicatif.
- Il y aura moins d'impact sur les tests fonctionnels.

➤ Bonne pratique de code

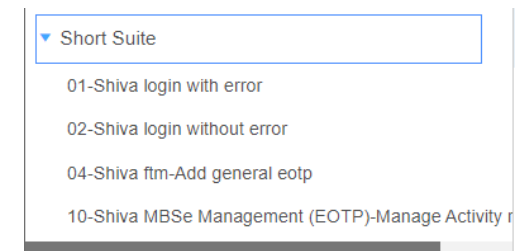
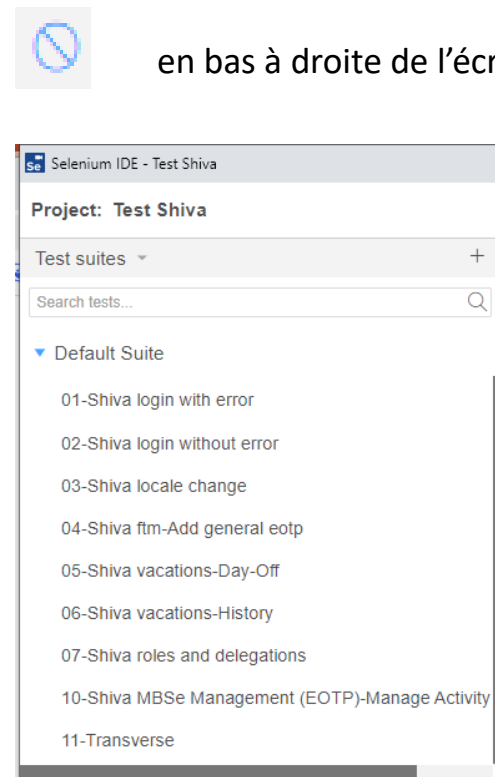
- Ajoutez des commentaires sur des groupes d'étapes. Ils sont précédés de `//` dans le champ commande.

➤ Logs d'exécution

- Videz régulièrement les logs d'exécution, en cliquant sur  en bas à droite de l'écran.

➤ Suites de tests

- Utilisez des suites de tests
 - ✓ pour ordonner votre travail
 - ✓ pour ne pas tout exécuter systématiquement





Pour aller plus loin



RÉSULTAT EN LIGNE DE COMMANDE

Exécution de Selenium IDE en Ligne de commande

➤ Préconditions

- node (v8 ou 10)
- npm (NodeJS package manager) installé avec Node
- selenium-side-runner (Selenium IDE commande line runner)
- Driver de navigateur (pour une exécution en local)

➤ Commandes d'installation

- > brew install node
- > npm install -g selenium-side-runner
- > npm install -g nomDriver
chromedriver (Chrome),
edgedriver (Edge) ou
geckodriver (Firefox)

➤ Commandes d'exécution

- Lancer Chrome driver
 - > cd
C:\Users\CorinneVin\AppData\Roaming\npm\node_modules\chromedriver\lib\chromedriver
 - > chromedriver
- Lancer Selenium-side-runner sur le ou les fichiers .side
 - > selenium-side-runner /chemin/projet/*.side

➤ URL

- <https://www.selenium.dev/selenium-ide/docs/en/introduction/command-line-runner>

Lancement de Chrome driver

```
C:\WINDOWS\system32\cmd.exe
Users\CorinneVin\AppData\Roaming\npm\node_modules\chromedriver\lib\chromedriver
C:\Users\CorinneVin\AppData\Roaming\npm\node_modules\chromedriver\lib\chromedriver
r>chromedriver
Starting ChromeDriver 88.0.4324.96 (68dba2d8a0b149a1d3afac56fa74648032bcf46b-refs
/branch-heads/4324@{#1784}) on port 9515
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggesti
ons on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

Lancement de Selenium-side-runner

```
Sélection Invite de commandes
Microsoft Windows [version 10.0.19041.804]
(c) 2020 Microsoft Corporation. Tous droits réservés.

C:\Users\CorinneVin>selenium-side-runner --base-url https://shiva.viseo.net/ --server http://localhost:9515 -c "browserName=chrome" C:/CorinneVin/01-Missions/00-TestsFonctionnelsSurShiva_SeleniumIDE/TestShiva_20210303_Chrome.side
info: Running C:/CorinneVin/01-Missions/00-TestsFonctionnelsSurShiva_SeleniumIDE/TestShiva_20210303_Chrome.side
FAIL ./DefaultSuite.test.js (106.651s)
Default Suite
  ✓ 01-Shiva login with error (2297ms)
  x 02-Shiva login without error (2789ms)
  x 03-Shiva locale change (17028ms)
  x 04-Shiva ftm-Add general eotp (17030ms)
  x 05-Shiva vacations-Day-Off (16853ms)
  x 06-Shiva vacations-History (16949ms)
  ✓ 07-Shiva roles and delegations (12343ms)
  x 10-Shiva MBSe Management (EOTP)-Manage Activity reports (15592ms)
  x 11-Transverse (3164ms)

Test Suites: 1 failed, 1 total
Tests: 7 failed, 2 passed, 9 total
Snapshots: 0 total
Time: 106.849s, estimated 137s
Ran all test suites.

C:\Users\CorinneVin>
```

EXPORT DE FICHIER

Une limite pour Selenium IDE ?

λ Test incontournable

λ Tester à minima

- le bon nom d'un fichier exporté
- sa bonne extension
- son emplacement

C'est possible avec la commande open file:///emplacement/nom du fichier + son extension

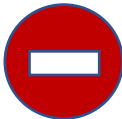
Command	Target	Value
open	file:///C:/Users/CorinneVin/Downloads/time-sheets-Imputations_Excess_billing_Sup_hours_liste_demo.xlsx	



λ Tester aussi

- sa taille (>0)
- son contenu

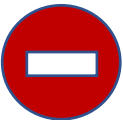
Impossible



Test à faire manuellement
ou à automatiser par un autre moyen

IMPORT DE FICHIER

Impossible




Test à faire manuellement
ou à automatiser par un autre moyen

RACCOURCIS CLAVIER


Comportement inattendu

Exemple simple


Command	Target	Value		Fonctionne
Send keys	Id=password	\${KEY_ENTER}		

Combinaison de touches

- Ex: Rafraichissement de l'écran
✓ ctrl F5

Command	Target	Value		1 ^{er} ctrl : appuyé 2 ^{ème} ctrl : relâché
Send keys	Id=password	\${KEY_CONTROL}\${KEY_F5}\${KEY_CONTROL}		

- Ex: Tabulation
✓ tab / shift tab

Command	Target	Value		1 ^{er} shift : appuyé 2 ^{ème} shift : relâché
Send keys	Id=password	\${KEY_SHIFT}		
Send keys	Id=password	\${KEY_SHIFT}\${KEY_TAB}\${KEY_SHIFT}		

Exécution

- Côté Selenium IDE :
✓ Raccourcis et tests OK
- Côté Navigateur Web :
X Commandes non exécutées lors du replay

Pourquoi ?

- Certains browsers interceptent l'événement lié à l'utilisation de certaines touches du clavier, avant d'avoir atteint la cible.
- Automatiser l'utilisation de ces touches n'est pas comme actionner physiquement ces touches.

Conclusion

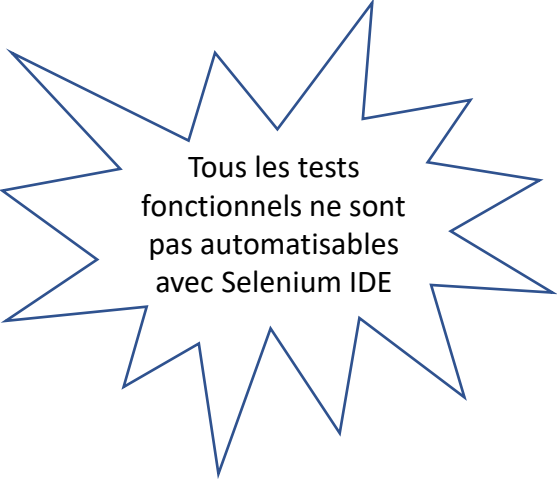
- Les raccourcis claviers fonctionnent dans Selenium IDE : les tests associés avant/après ces commandes sont concluants.
- Inutile donc de voir les actions correspondantes se jouer dans le navigateur Web lors du replay des tests.

INCONVÉNIENTS / AVANTAGES

Selenium IDE

➤ Inconvénients

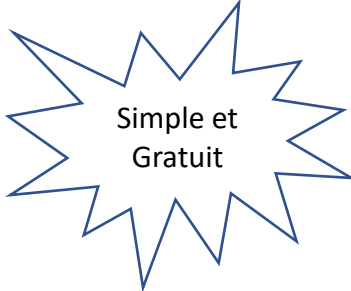
- Extensions Firefox et Chrome
 - ✗ Pas officiellement sur d'autres navigateurs
- Uniquement les applications Web
 - ✗ Pas d'applications mobiles
- Bases de données
 - ✗ Pas de Tests BDD
- Import/export de fichiers non gérés
 - ✗ Pas d'accès au contenu de fichiers
- Mots de passe
 - ✗ stockés en texte brut



Tous les tests
fonctionnels ne sont
pas automatisables
avec Selenium IDE

➤ Avantages

- Installation et utilisation simples
- Aucune connaissance en programmation requise
- Aide au débogage grâce aux messages d'erreurs, aux points d'arrêts
- Possibilité d'ajout manuel
 - ✓ de commandes (par exemple : vérifications, etc.)
 - ✓ de commentaires
- Export de tests Selenium IDE
 - ✓ en nombreux langages,
 - ✓ permettant leur exécution ailleurs que dans Selenium IDE
- Automatisation de tests fonctionnels de + en + compliqués
- Gratuit



Simple et
Gratuit

LIENS UTILES SUR SELENIUM IDE

› Sites officiels

- <http://seleniumide.org/>
- <https://www.selenium.dev/selenium-ide/>

› Commandes Selenium IDE

- <https://www.seleniumhq.org/selenium-ide/docs/en/api/commands/>

› Contrôles (if, while etc.)

- <https://www.selenium.dev/selenium-ide/docs/en/introduction/control-flow>

› Documentation - Créer et tester des flux Selenium IDE

- <https://docs.microsoft.com/fr-fr/power-automate/desktop-flows/create-web>

› Exécution de Selenium IDE en ligne de commande

- <https://www.selenium.dev/selenium-ide/docs/en/introduction/command-line-runner>

› Scripts

- <https://ui.vision/rpa/docs/selenium-ide/executescript>

Merci!

- un ROTI ?
 - ✓ Tapez un chiffre dans la discussion teams.



The background is a dark blue gradient with a bokeh effect of out-of-focus city lights in various colors (blue, yellow, orange, pink). Two large, white, diagonal geometric shapes, resembling stylized chevrons or arrows, are positioned on the left side of the image.

Corinne VIN

Analyste fonctionnel / Intégrateur sénior

WISEO LYON

+33 (0)4 72 33 78 30

Corinne.Vin@Viseo.com