

Logs vulcain dispatcher

- dans l'interface web en cliquant sur le produit
- requête MongoDB via une rails console sous vulcain-api

Exemples :

- `Log.where('session.uuid' => '')`
- `Log.where('session.uuid' => ', :verb => 'assess') ...` cf doc mongodb

Fichier syslog sous `/var/log/vulcain-dispatcher/vulcain-dispatcher.log`

Logs vulcain

En cas de crash d'un vulcain.

- Fichier syslog sous `/var/log/vulcain-dispatcher/vulcain.log`

Via logs robots.rb

Rappel : les stratégies sont exécutées sur les vulcains donc les numéros de ligne des logs sont les numéros de ligne de `vulcain/lib/core/robots.rb`

Si erreur, voir la stack trace dans la console et repérer les dernières lignes de passage dans `robots.rb`. Ensuite ouvrir `robots.rb` ...

Via les tests d'intégration

Tous les tests integrations des strategies héritent de `StrategyTest`. Ils sont sous `test/integration/robot`. Si une donnée du contexte est à modifier temporairement c'est ici qu'il faut la changer (`common_context`).

Toutes les stratégies on des tests

- "finalize order" => arrête avant remplissage paiement
- "validate order" => rempli info bancaire et valide paiement.

Il suffit de créer un test unitaire semblable à "finalize order", "validate order" ou un autre avec l'url du produit puis de le lancer et de voir où ça casse ... puis fixer :)

Notes:

Le robot est découpé en modules sous `robot/core` (actions, logout, registration

...) qui ont leur tâche propre. Ce n'est pas encore très propre car il reste des choses à faire. Mais généralement si un script casse pour la création d'un compte c'est dans Registration qu'il faut chercher l'erreur.

De même les identifiants des éléments sont sous le Vendor Constants (ie AmazonFranceConstants) dans le fichier <vendor>.rb. Un identifiant est soit un xpath(commençant obligatoirement par "//"), soit une chaîne de caractères;

Dans ce dernier cas le robot cherche soit un tag "a" avec texte la chaîne soit Voir la méthode **Driver#find_elements_with_pattern** pour comprendre ce qui est recherché quand on met une chaîne de caractères comme identifiant.

Lorsqu'il y a check:true comme paramètre, cela signifie que l'on avance si le xpath n'est pas présent dans le dictionnaire du Vendor, où si le xpath est dans le dico mais qu'aucun élément n'est trouvé ... sans check:true, le robot attend l'élément et lève une exception si non trouvé.