

Retour d'expérience tests Cypress et comparaison avec PHPUnit et Behat



Qui suis-je ?

- Florent Torregrosa (Grimreaper)
- Début d'utilisation avec Drupal 6 en 2011
- Tech expert Drupal chez Smile
- Contributions :
 - ex-membre du bureau de l'ADFF
 - contributions diverses pour Drupal France
 - co-organisateur d'événements
 - traductions
 - modules
 - patches



Sommaire

- 1) Qu'est-ce que Cypress ?
- 2) Mise en place
- 3) Présentation du code et démo
- 4) Comparaison avec PHPUnit et Behat



1 : Qu'est-ce que Cypress ?

- Framework de tests end-to-end : tests sur un environnement réel complet
- Écrit en Javascript
- Pilote un navigateur de manière fine
- Complètement indépendant de Drupal
- Console pour voir les tests s'exécuter et plus facilement les débbugger



1 : Qu'est-ce que Cypress ?

- Comparé à Behat :
 - Les tests ne sont pas écrits en Gherkin
 - Nécessite un environnement NodeJs pour fonctionner
 - Beaucoup moins de couches d'abstraction nécessaires pour piloter le navigateur et simple de passer d'un navigateur à l'autre
 - Plus simple à débbugger
 - Pas de raccourci d'implémentation comme avec l'extension Behat pour Drupal



1 : Qu'est-ce que Cypress ?

- Comparé à PHPUnit :
 - Tests plus rapides car s'exécutant sur un environnement existant
 - Modifie l'environnement de test
 - Si pas de retour d'actions dans le navigateur (page BO à consulter ou message), pas moyen de lancer des vérifications de valeurs sauvegarder ou autres sans y accéder comme un admin le ferait



2 : Mise en place

- Possibilité d'utiliser un module Cypress qui va permettre de lancer Cypress via Drush et de lancer les tests sur un environnement éphémère
- Mise en cache de la création de l'environnement de test pour accélérer le setup entre chaque tests et avoir des tests indépendants



2 : Mise en place

- Mais... cela suppose que NodeJS et Cypress sont installés sur le même environnement que PHP
- Or utilisant des environnements Docker, avec un conteneur NodeJS séparé et afin d'éviter des problèmes de librairies, etc et devoir maintenir cela, je souhaite pouvoir utiliser les images Docker officielles Cypress



2 : Mise en place

- 3 types d'images officielles :
 - cypress/base : NodeJS + dépendances
 - cypress/browsers : base + navigateurs (reste à installer la version de Cypress souhaitées)
 - cypress/included : browsers + Cypress
- Utilisation de cypress/included par souci de facilité



3 : Présentation du code et démo

- La mise en place Cypress a été faite dans un premier temps
- L'écriture des tests dans un second temps, l'écriture a pris une longue après-midi



4 : Comparaison (subjective)

Points de comparaison	PHPUnit	Behat	Cypress
Installation	Core + image officielle pour JS	Extension Behat Drupal	Complexe
Environnement	PHP	PHP	NodeJS
Débogage	Normal	Complexe	Facile/Normal
Accès API Drupal	Complet	Partiel (via extension)	Aucun (sauf à utiliser le module)
Environnement de test	Créé puis détruit	Existant	Existant
Intégration dans une CI	Ok	Ok	Pas encore



Conclusion

- Pour du tests sur l'application finale, Cypress est très adapté
- Complémente la panoplie d'outils de tests, à utiliser lorsqu'il est pertinent
- Compliqué à mettre en place dans une CI, à moins d'avoir des images Docker dédiées par projet



Merci pour votre attention

Avez-vous des questions ?



Liens

- <https://www.cypress.io/>
- <https://docs.cypress.io/plugins/directory>
- <https://docs.cypress.io/api/table-of-contents>
- <https://github.com/cypress-io/cypress-docker-images>
- <https://www.drupal.org/project/cypress>

