

La réalisation d'un test fonctionnel

Table des matières

I. Réalisation d'un test fonctionnel	3
II. Exercice : Partie 1 : les bases du test fonctionnel	3
III. Exercice : Partie 2 : aller plus loin sur les tests fonctionnels avec Symfony	3
Solutions des exercices	3

I. Réalisation d'un test fonctionnel

Conseil

Cette application permet de valider vos connaissances et acquisitions de la compétence, en lien avec le référentiel du parcours visé. Elle ne participe pas à l'obtention de votre parcours, mais vous permet de vous entraîner à l'appropriation de la pratique professionnelle attendue.

Veillez à la qualité de votre syntaxe.

Durée indicative : 1 heure

Documents autorisés : tous les documents sont autorisés

Matériel autorisé : tous les matériels sont autorisés

II. Exercice : Partie 1 : les bases du test fonctionnel

En entreprise, on vous demande de travailler sur un nouveau projet. Afin d'appréhender celui-ci, vous allez tout d'abord découvrir l'ensemble des pages web du site. Idéalement les tests fonctionnels sont déjà écrits, vous possédez donc une documentation précise. Mais dans le cas contraire, c'est votre rôle de réaliser les tests fonctionnels. Nous verrons ici comment écrire ces tests sur une application Symfony.

Question 1

[solution n°1 p.5]

1.1. Qu'est-ce qu'un test fonctionnel ?

Question 2

[solution n°2 p.5]

1.2. Comment mettre en place un test fonctionnel ?

Question 3

[solution n°3 p.5]

1.3. Comment peut-on créer un test fonctionnel ?

III. Exercice : Partie 2 : aller plus loin sur les tests fonctionnels avec Symfony

Vous assumerez le rôle d'un développeur qui découvre de nouvelles applications. Votre tâche consiste à mettre en œuvre des tests fonctionnels pour documenter l'ensemble de l'application. Vous êtes dans le cas où vous ne connaissez pas encore ces fonctionnalités, et ne savez pas à quoi ressemblent les pages du site.

Question 1

[solution n°4 p.5]

2.1. Comment installer Symfony pour réaliser des tests fonctionnels ?

Question 2

[solution n°5 p.5]

2.2. Comment tester et soumettre ses tests fonctionnels ?

Question 3

[solution n°6 p.6]

2.3. Quels sont les autres outils utiles pour l'utilisation des tests fonctionnels ?

Solutions des exercices

p. 3 Solution n°1

Le test fonctionnel permet de tester automatiquement toutes les fonctionnalités d'une application. Cela comprend entre autres les fonctionnalités demandées dans le cahier des charges du projet. Grâce au test fonctionnel, il est possible de tester si un membre a bien la possibilité de s'inscrire, de se connecter, de se déconnecter, etc.

De plus, dans le cadre de ces tests, il est possible d'ajouter un scénario d'utilisation afin de parvenir au résultat souhaité. En effet, la réalisation d'un scénario vous permettra d'imiter le comportement que pourrait avoir un utilisateur lorsqu'il teste l'application, pour ensuite régler les problèmes potentiels.

p. 3 Solution n°2

Les tests fonctionnels se font sur des applications déjà développées. Il y a plusieurs étapes pour effectuer un test fonctionnel :

- Dans un premier temps, il faut créer un client HTTP,
- Ensuite, envoyer une requête HTTP grâce au client HTTP sur une page à tester,
- Après avoir récupéré la réponse HTTP, vous devez regarder si le contenu de la réponse était bien celui attendu.

Les tests fonctionnels peuvent également être ajoutés à différents moments du projet, par exemple avant ou après l'implémentation d'une nouvelle fonctionnalité ou alors à la découverte d'une nouvelle anomalie.

p. 3 Solution n°3

Durant l'écriture d'un test fonctionnel, nous effectuons des requêtes vers une URL (la plupart du temps nommée « route »), pour récupérer ce qu'elle renvoie et ensuite déterminer s'il y a des anomalies. Le point d'entrée de l'application se situe au niveau du contrôleur qui correspond à son URL. Ensuite, vos classes de tests auront pour obligation de récupérer l'arborescence des classes de contrôleur liées aux URL testées. Cependant, avant de commencer vos tests, il vous faut faire la liste de toutes les routes potentielles, pour ainsi évaluer la quantité de travail à effectuer.

p. 3 Solution n°4

Pour commencer, il vous faut un projet Symfony fonctionnel. Une fois le projet Symfony installé et lancé sur votre ordinateur, vous pouvez à présent configurer le projet Symfony, ceci vous permet de faire vos premiers tests fonctionnels.

La configuration se fait sur un invite de commande, en se positionnant à la racine du projet Symfony, pour ainsi effectuer ces trois commandes :

1. composer install
2. bin/console doctrine:database:create --env=test
3. bin/console doctrine:schema:create --env=test

Au moment où ces commandes auront été effectuées, vous pourrez ensuite écrire des tests fonctionnels.

p. 3 Solution n°5

Pour effectuer cette tâche, nous allons faire appel au crawler. Le crawler est un objet Symfony auquel vous avez accès après chaque requête, voici comment l'utiliser :

```
1 <?php
2 namespace Tests\AppBundle\Controller;
3 use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
4 class DiaryControllerTest extends WebTestCase
5 {
6     // ...
7
8     public function testAccueil()
9     {
10         $testClient = static::createClient();
11         $testCrawler = $testClient -> request('GET', '/');
12
13         $this->assertSame(1, $testCrawler -> filter('html:contains("Bonjour et bienvenue.")')-
14 >count());
15 }
```

Le crawler est une instance de la classe `Symfony\Component\DomCrawler\Crawler`.

Ici, nous regardons le contenu de la page, l'instruction `$testCrawler -> filter('html:contains("Bonjour et bienvenue.")')->count();` permet de configurer le nombre d'éléments de la balise html avec le texte « *Bonjour et bienvenue.* ».

p.3 Solution n°6

Les tests fonctionnels sont le plus souvent utilisés avec des applications Symfony, mais il existe également d'autres outils permettant de faire ces tests. Cela peut même être bénéfique, car il est possible de les cumuler.

Dans un premier temps, il y a la librairie Behat qui priorise une méthode nommée BDD (Behaviour Development Driven) qui favorise les choix du développement côté business. Cette manière d'écrire les tests respecte au maximum les scénarios de départ.

Il y a également la librairie PHPSpec, qui facilite le TDD (Test Driven Development), car pour chaque classe en développement, cette librairie vous demande de décrire son comportement.

En somme, il existe encore une multitude d'outils. De plus, il est primordial de faire attention aux maintenances de chaque test ; cela prendra un certain temps qu'il ne faut pas négliger.