



REUNION D'INFORMATION PRF RÉGION OCCITANIE

Guivez-nous.. in

www.linkedin.com/school/adrarnumerique





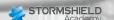
- > INFRASTRUCTURES SYSTÈMES & RÉSEAUX
- > CYBERSÉCURITÉ INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > TRANSFORMATION NUMERIQUE DES ENTREPRISES

www.adrar-numerique.com



















Les tests fonctionnels







GEN GRANDE ÉCOLE DU NUMÉRIQUE











Les tests unitaires (1/6) Symfony



Les tests unitaires sont des procédures qui permettent de vérifier le bon fonctionnement d'une partie précise d'un programme qu'on appel unité.

Il vous faudra autant de tests que de cas de sortie dans vos méthodes (sur le papier)...

Dans la pratique, nous ne cherchons pas à couvrir 100% de nos applications avec les tests, ils permettent avant tout de tester du code sensible, qui peut impliquer plusieurs choses...

Pour faire ces tests vous devez télécharger ces deux dépendances:

 $\$ \ \ \text{composer require --dev phpunit/phpunit \&\& composer require --dev symfony/phpunit-bridge}$

Pour créer un test, utilisez la commande suivante:

\$ symfony console make:test

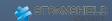
Pour lancer les tests (qui se trouvent tous dans le dossier "tests" à la racine du projet), utilisez la commande suivante:

\$ vendor/bin/phpunit



















Les tests unitaires (2/6) Symfony



estCase

Prenons l'exemple avec mon entité Formateur

```
public function testMdpGettersAndSetters()
                                                                                                         FormateurTest.php
   $formateur = new Formateur();
   $this->assertNull($formateur->getMdp());
    $formateur->setMdp('Severity@Viselike9@Clobber');
    $this->assertEquals('Severity@Viselike9@Clobber', $formateur->getMdp());
public function testAddCour()
   $formateur = new Formateur();
   $cours = new Cours();
    $formateur->addCour($cours);
   $this->assertCount(1, $formateur->getCours());
   $this->assertTrue($formateur->getCours()->contains($cours));
```

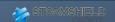
lci je viens vérifier le bon fonctionnement de l'accesseur et du mutateur de ma propriété "mdp"

Ensuite, je test l'ajout d'un cours

















GEN



Les tests unitaires (3/6) Symfony



Il est possible de passer un ensemble de données à une même méthode d'un coup

```
ExempleTest.php
estCase
           * @dataProvider additionProvider
         public function testAdd($a, $b, $expected)
              $this->assertEquals($expected, $a + $b);
         public static function additionProvider()
             return [
                  [0, 0, 0],
                  [0, 1, 1]
              ];
```















Les tests unitaires (4/6) Symfony



estCas

Prenons l'exemple avec mon entité Formateur

```
public function testPresenceId(): void

{
    $response = static::createClient()->request('GET', 'http://api.marceau-rodrigues.fr/api');

    $this->assertResponseIsSuccessful();
    $this->assertJsonContains(['@id' => '/api']);
}
```

lci, je viens vérifier que ma route /api contient bien une clé "@id" et une valeur "/api"





















Les tests unitaires (5/6)

Symfony

nerTes ase

Prenons l'exemple avec mon entité Formateur

```
public function testSomething(): void

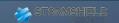
{
    $client = static::createPantherClient();
    $crawler = $client->request('GET', '/');

$this->assertSelectorTextContains('h1', 'Hello World');
}
```















Les tests unitaires (6/6) Symfony



Les retours des tests unitaires

Time: 00:00.125, Memory: 24.00 MB

OK (1 test, 2 assertions)

WARNINGS!

Tests: 1, Assertions: 2, Warnings: 2.

FAILURES!

Tests: 1, Assertions: 2, Failures: 1.

Lancez la commande qui suit pour voir le taux de couverture de votre code (entity/controller/repository/datafixtures...)























