

- Les mocks

- PLAN

- 1 - Exercice sur les tests unitaires
- 2 - Le concept de mock
- 3 - Mise en pratique

1

EXERCICE SUR LES TESTS UNITAIRES

EXERCICE : la classe “User”

- Implémenter une classe User et sa classe de test
- Détails fonctionnels :
 - Un User possède un email, un nom, un prénom, un âge
 - Un User est valide si son email est renseigné et a le bon format, ses nom et prénom sont renseignés et si il a plus de 13 ans
- Opérations :
 - isValid() dans la classe User
 - testIsValid() dans la classe UserTest



EXERCICE : la classe “Product”

- Implémenter une classe Product et sa classe de test
- Détails fonctionnels :
 - Un Product possède un nom et un owner (de type User)
 - Un Product est valide si son nom est renseigné si son owner est renseigné et valide
- Opérations :
 - isValid() dans la classe Product
 - testIsValid() dans la classe ProductTest





2

LE CONCEPT DE MOCK

● LES MOCKS

- ◦ Objet “magique” au comportement pré-câblé
- Permet de configurer le comportement de l’objet sous la forme :

*Quand on te demande ça,
tu fais ça !*

● LES AVANTAGES

- ◦ Ne pas s'occuper des dépendances du SUT*
- Maîtriser le comportement d'un autre objet
- Tester facilement des cas d'erreurs
- Tester lorsque des dépendances ne sont pas encore développées

*System Under Control

- A QUOI RESSEMBLE UN MOCK ?

- ◦ Création du mock :

```
$obj = $this->createMock(SomeClass::class);
```

- Utilisation du mock :

```
$obj->expects($this->any())  
    ->method('doSomething')  
    ->will($this->returnValue(42));
```



3

MISE EN PRATIQUE

EXERCICE : la classe “Exchange”

- Implémenter une classe Exchange et sa classe de test
- Détails fonctionnels :
 - Un Exchange possède un **receiver** (de type User), un **produit** (de type Product), un **owner** contenu dans le produit (de type User), une **date de début**, une **date de fin**, un composant d'envoi d'email (de type **EmailSender**) et un composant de connexion à la base de données (de type **DBConnection**)
 - L'échange est enregistré si le owner, le receiver et le produit sont valides. Il faut que les dates soient valides (date de début dans le future)
 - Si le receiver est mineur, lui envoyer un mail
- Opérations :
 - save() dans la classe Exchange
 - testSave() dans la classe ExchangeTest

