

Les mocks

¥ Pré-requis :

- ! Eclipse (ou IDE équivalent) installé
- ! JDK 8+ installé
- ! Maven 3.6.3+ installé (facultatif)

Import Projet

1. Démarrer Eclipse
2. Importer le projet `mockito-practical-work` comme un projet Maven (menu contextuel *Import* → *Existing Maven projects* dans la vue *Package Explorer*)

Le composant ^ tester

¥ La classe `Order` représente une commande comportant une liste de produits (voir classe `Product`).

Mes premiers Mocks

1. Dans l'explorateur, faites un clic droit sur la classe `Order` (sous le package `com.sqli.isc.iut.courses.mockito.domain`).
2. Dans le menu contextuel, cliquez sur *New* > *JUnit Test Case*.
3. Sélectionnez le bouton radio *New JUnit 4 test* (sélectionné par défaut).
4. Remarquez que le dossier *Source folder* pointe sur le répertoire `mockito-practical-work/src/test/java`, répertoire standard en Java où sont placés les tests unitaires.
5. Cliquez sur le bouton *Finish*.
6. Nous allons donc écrire un test simple qui permet de vérifier la création d'une nouvelle commande avec deux nouveaux produits.

La règle est la suivante : si j'ai deux produits dont les prix sont 3.99 et 5.00, ma commande me revient ^ 8.99

La librairie [Mockito](#) propose plusieurs méthodes pour instancier des Mocks. La façon la plus simple est d'utiliser la méthode statique `mock()` :

```
Mockito.mock(Product.class);
```

Pour simuler le comportement d'un produit pour qu'il retourne un prix concret et donc tester notre méthode `getTotalPrice()` :

```
Mockito.when(product1.getPrice()).thenReturn(new BigDecimal("3.99"));
```