25/04/2023
Version 4

SPRING

JÉRÉMY PERROUAULT

SPRING TEST

SPRING TEST AVEC JUNIT5

PRÉSENTATION DE SPRING TEST

- Un Test Unitaire permet de vérifier le bon déroulement d'une fonctionnalité
- Il doit être atomique
  - Léger
  - Isolé
  - Rapide

PRÉSENTATION DE SPRING TEST

- **Spring** permet de réaliser des tests unitaires
  - Reproductibles
  - Compréhensibles

- Combiné à un Framework de test, tel que **JUnit**

PRÉSENTATION DE SPRING TEST

- Création d'un répertoire *src/test* au même titre que *src/main*
  - Dans **Java Resources**
- Les packages restent identiques
- On ajoute `Test` à la fin du nom de la classe qu'on veut tester
- Exemple :
  - Tester la classe `fr.formation.repo.ProduitRepositorySpring`
  - ➔ Classe `fr.formation.repo.ProduitRepositorySpringTest` dans *test*

PRÉSENTATION DE SPRING TEST

- Création d'un répertoire *test/resources* au même titre que *main/resources*
  - Dans **Java Resources**
- Tous nos fichiers de configuration seront recopiés ici
  - S'il y a besoin de les adapter pour les tests (base de données différente par exemple)

PRÉSENTATION DE SPRING TEST

- Ces annotations sont à placer sur les méthodes de la classe de test
  - Ces annotations proviennent de **JUnit**

Annotation	Définition
@Test	Méthode de test qui sera exécutée
@BeforeAll	Méthode statique qui sera exécutée avant le premier test
@AfterAll	Méthode statique qui sera exécutée après le dernier test
@BeforeEach	Méthode qui sera exécutée avant chaque test
@AfterEach	Méthode qui sera exécutée après chaque test

PRÉSENTATION DE SPRING TEST

- Import du package `org.junit.Assert.*`

Assertion	Description	Déclenchement
fail	Echec	Toujours
assertEquals	Vérifier une égalité	Si différent
assertTrue	Vérifier une condition	Si la condition est fausse
assertNotEquals	Vérifier une non-égalité	Si équivalent
assertFalse	Vérifier une non-condition	Si la condition est vraie
assertNull	Vérifier si l'objet est *null*	Si l'objet n'est pas *null*
assertNotNull	Vérifier si l'objet n'est pas *null*	Si l'objet est *null*

CONFIGURATION

CONFIGURATION DE SPRING TEST

CONFIGURATION

- Ajout de la dépendance **spring-test**
  - Avec le scope *test*

```
<!-- Spring Test -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>${spring.version}</version>
  <scope>test</scope>
</dependency>
```

CONFIGURATION

- Ajout des dépendances **JUnit** & **Mockito**

```
<!-- JUnit -->
<dependency>
  <groupId> org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>5.9.2</version>
  <scope>test</scope>
</dependency>
```

```
<!– Mockito -->
<dependency>
  <groupId>org.mockito</groupId>
  <artifactId>mockito-junit-jupiter</artifactId>
  <version>5.3.1</version>
  <scope>test</scope>
</dependency>
```

CONFIGURATION

- Classe de test annotée
  - `@SpringJUnitConfig(AppConfig.class)`
    - `@SpringJUnitWebConfig` (sera utilisée pour retrouver le contexte d'application Web)
- ET / OU
  - `@ExtendWith(MockitoExtension.class)`
    - Pour exécuter le test avec **Mockito** et injecter des **mocks**
    - `@InjectMocks`
      - Injecter des composants **Spring** simulés
    - `@Mock`
      - Simuler un composant **Spring**

EXERCICE

- Rédiger un test pour vérifier qu'un **bean** géré par **SPRING** existe bien dans son contexte
  - Vérifier que le **Guitariste** existe bien