



SPRING

Jérémy PERROUAULT



SPRING TEST

Introduction à
Spring Test avec JUnit

TEST UNITAIRE

Un Test Unitaire (TU) 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

ORGANISATION

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.dao.ProduitDAO
- ➔ Création d'une classe fr.formation.dao.ProduitDAOTest dans le répertoire « test »

ORGANISATION

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)

CONFIGURATION

Ajout de la dépendance Spring-Test

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

CONFIGURATION (JUNIT)

Ajout de la dépendance JUnit & Mockito

```
<!-- JUnit -->  
<dependency>  
  <groupId> org.junit.jupiter</groupId>  
  <artifactId> junit-jupiter-api</artifactId>  
  <version>5.8.2</version>  
  <scope>test</scope>  
</dependency>
```

```
<!-- Mockito -->  
<dependency>  
  <groupId>org.mockito</groupId>  
  <artifactId>mockito-junit-jupiter</artifactId>  
  <version>4.3.1</version>  
  <scope>test</scope>  
</dependency>
```


CONFIGURATION (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

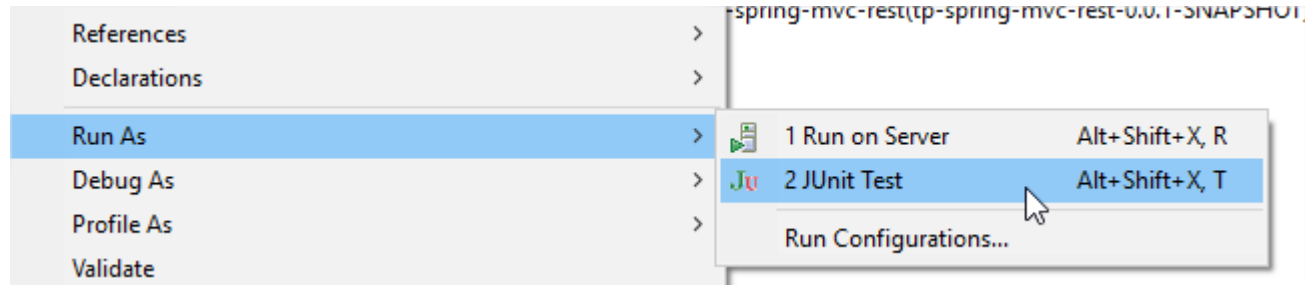
ASSERTION

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

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

EXÉCUTION (JUNIT)

Sélection de JUnit Test



CONFIGURATION (JUNIT 5)

Classe de test annotée

- **@SpringJUnitConfig(AppConfig.class)**
 - **@SpringJUnitWebConfig** (sera utilisée pour retrouver le contexte d'application Web)

ET / OU

- **@ExtendsWith(MockitoExtension.class)**
 - Pour exécuter le test avec Mockito et injecter des mocks
- **@InjectMocks**
 - Injecter des composants Spring simulés
- **@Mock**
 - Simuler un composant Spring

CONFIGURATION (JUNIT 4)

Classe de test annotée

- **@RunWith(SpringRunner.class)**
 - @RunWith(SpringJUnit4ClassRunner.class) également possible
- **@ContextConfiguration(classes = AppConfig.class)**
- **@WebAppConfiguration** dans un contexte Web

EXERCICE

Rédiger un test pour vérifier qu'un bean géré par SPRING existe bien dans son contexte

- Vérifier que le Guitariste existe bien