

TP 01

SPRING – BEANS XML

PROJET CONSOLE ADMIN












- L'objectif de cette section est de migrer le projet « pizzeria-console » sous Spring.
- Le projet Spring est constitué de nombreux sous-projets avec chacun son cycle de vie. Pour éviter d'utiliser deux sous-projets avec des versions incompatibles, le projet met à disposition un BOM (Bill Of Materials). Il s'agit d'une dépendance Maven spéciale à intégrer avec le scope « import ». Compléter le projet parent, dans la section « dependencyManagement » avec :

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-framework-bom</artifactId>
  <version>4.2.6.RELEASE</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

- Pour créer un contexte Spring dans le projet « pizzeria-console », ajouter la dépendance vers « spring-context ».

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
</dependency>
```

Voici la hiérarchie des dépendances :

- ▼  spring-context : 4.2.6.RELEASE [compile]
 - ▼  spring-aop : 4.2.6.RELEASE [compile]
 -  aopalliance : 1.0 [compile]
 -  spring-beans : 4.2.6.RELEASE (omitted for conflict with
 -  spring-core : 4.2.6.RELEASE (omitted for conflict with
 - ▼  spring-beans : 4.2.6.RELEASE [compile]
 -  spring-core : 4.2.6.RELEASE (omitted for conflict with
 - ▼  spring-core : 4.2.6.RELEASE [compile]
 -  commons-logging : 1.2 [compile]
 - ▼  spring-expression : 4.2.6.RELEASE [compile]
 -  spring-core : 4.2.6.RELEASE (omitted for conflict with

- Le contexte Spring va être construit à partir de deux fichiers :
 - dao-memoire-config.xml qui définit les beans :
 - daoFactory
 - pizzaDao (implémentation mémoire)
 - application-config.xml qui va définir les beans :
 - menuPrincipal
 - scanner
 - Pour injecter un appel à une méthode statique, vous pouvez utiliser `<util:constant>`. Exemple :

```
<constructor-arg>
  <util:constant static-field="java.lang.System.in" />
</constructor-arg>
```

- Désormais le pilotage du choix de l'implémentation du DAO à utiliser se fera via la définition du fichier spring à utiliser dans le fichier « application.properties ». Exemple de contenu :

dao.impl=dao-memoire-config.xml

- Cette configuration remplace la précédente dirigée par des entiers.
- Compléter la classe « PizzaAdminApp » pour créer un contexte Spring, récupérer le bean « menuPrincipal » puis appeler sa méthode « afficher »
- A cette étape, l'implémentation mémoire fonctionne.

ACCES A LA BASE DE DONNEES

- Créer les configurations liées aux implémentations :
 - JDBC (dao-jdbc-config.xml)

- JPA (dao-jpa-config.xml)

- Pour créer un bean à partir de l'exécution d'une méthode statique, utiliser l'attribut « factory-method »

```
<bean id="bean1" class="fr.pizzeria.MaClasse" factory-  
method="maMethodeStatique">  
  <constructor-arg value="parametreMethode" />  
</bean>
```

- Etc...

- Vérifier le fonctionnement de l'application dans ces différentes configurations.