



02 Spring Core - 03configuration basée sur des Annotations

Annotations : définitions

- Les annotations en java sont des marqueurs, des étiquettes placées sur des classes.
- Elles fournissent des méta-données au sujet de la classe
- Elles sont prises en compte au moment de la compilation, ou de l'exécution du code pour des traitements particuliers.
-

Activer le Scan des packages

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">

  <context:component-scan base-package="com.springdemo"/>

</beans>
```

À partir de ce chemin de base vers un package, au démarrage, spring va scanner les fichiers qu'il contient récursivement, à la recherche d'annotations.

Spring va alors considérer les annotations et réaliser les traitements appropriés à la compilation ou à l'exécution.

Par exemple, spring peut enregistrer un bean, si l'annotation posée sur la classe est `@Component`.

TD – configurer des beans par annotation

- Modifier le fichier de configuration xml
- Ajouter des @Component sur vos classes d'implémentation
- A partir d'une méthode main accédez au contexte pour récupérer ces beans.

file: applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="com.springdemo"/>

</beans>
```

```
package com.springdemo;

public interface Musicien{
    public String      joueTaPartition();
}
```

```
package com.springdemo;

@Component("Banda Taurina")
public class Trompetiste implements Musicien {

    @Override
    public String joueTaPartition() {
        return "la trompette du toril";
    }
}
```

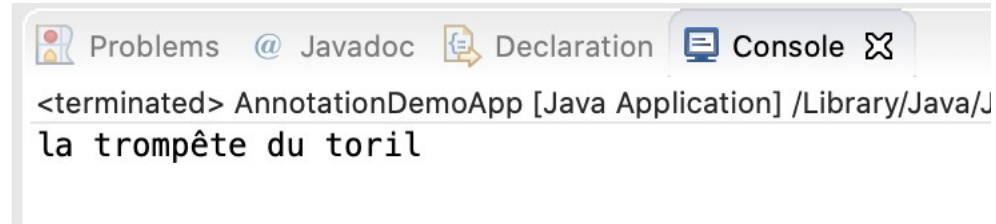
```
public class AnnotationDemoApp {

    public static void main(String[] args) {
        //charger le fichier de configuration xml
        ClassPathXmlApplicationContext context =
            new ClassPathXmlApplicationContext("applicationContext.xml");

        // accéder au bean géré par le spring container
        Musicien musicien= context.getBean("Banda Taurina", Musicien.class);
        System.out.println(musicien.joueTaPartition());

        context.close();
    }
}
```

Run As> ...



L'id par défaut de @Component

- Le paramètre de cette annotation est optionnel
- Pour la classe Trompetiste, les deux annotations suivantes sont équivalentes :

@Component("trompetiste") et @Component()

- Cela signifie que vous pouvez choisir un id pour ce bean et le spécifier en paramètre

@Component("Banda Taurina")

Il vous appartient ensuite d'appeler ce bean correctement pour y accéder dans le contexte

- Par défaut spring attribut un id basé sur les convention de nommage de la classe