



Spring Core – annotation

Bean scope

Scope – rappel de la définition

- En français = portée
- C'est un qualificatif de la manière dont Spring va instancier, "exposer" (on parle ici de visibilité), puis détruire chaque Bean.
- On peut dire que la notion de scope se réfère à celle de cycle de vie d'un bean
- Pour chaque bean , on peut spécifier une portée.
- Chaque bean possède une valeur de portée dite "par défaut" avec spring.

Singleton - rappel définition

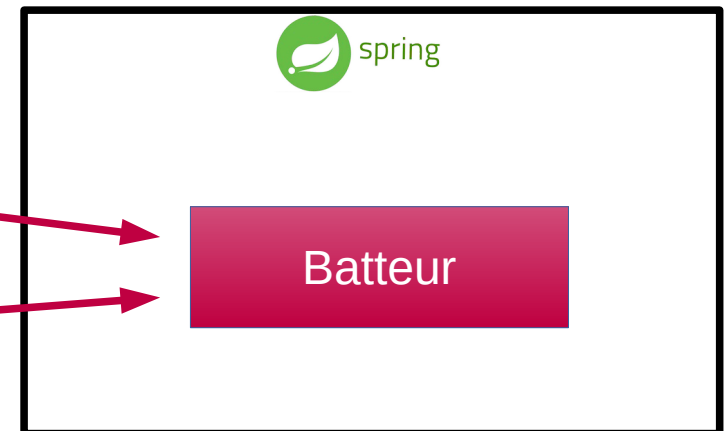
Quand le scope d'un bean est configuré explicitement (ou par défaut) comme un singleton, Spring ne va créer qu'une seule instance de ce bean.

Cette instance sera maintenue en cache

Tous les appels à ce bean accèderont à cette même instance.

File: demoApp.java

```
Musicien musicienA =  
    context.getBean("unMusicien", Musicien.class);  
  
Musicien musicienB =  
    context.getBean("unMusicien", Musicien.class);
```



File: applicationContext.xml

```
<bean id="unMusicien" class="com.springdemo.Batteur" > ... </bean>
```

Scope = rappel prototype

Un nouvel objet est crée à chaque requête

Ex :

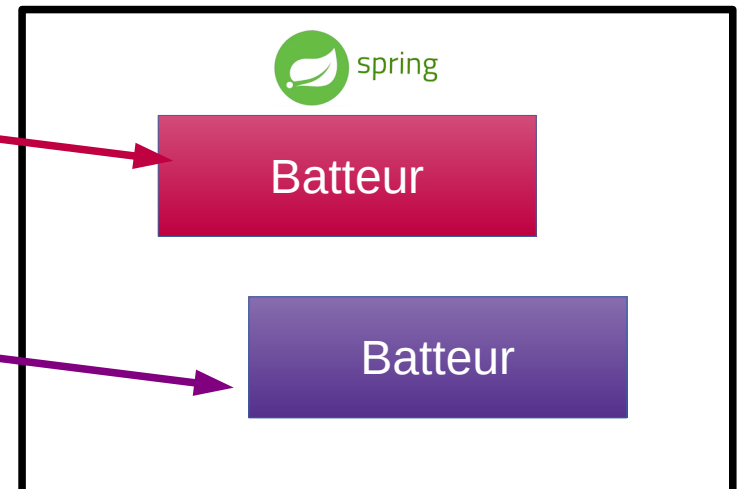
```
<beans . . . >

    <bean id="unMusicien"
          class="com.springdemo.Batteur"
          scope="prototype">
    </bean>

</beans>
```

```
Musicien musicienA =
    context.getBean("unMusicien", Musicien.class);
```

```
Musicien musicienB =
    context.getBean("unMusicien", Musicien.class);
```



Scope par annotation

```
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("singleton")
public class Trompetiste implements Musicien {
    . . .
}
```

```
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("prototype")
public class Trompetiste implements Musicien {
    . . .
}
```

Démo - TD

```
public class ScopeDemoApp {

    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
        Trompetiste a= context.getBean("trompetiste", Trompetiste.class);

        Trompetiste b= context.getBean("trompetiste", Trompetiste.class);

        // utiliser les methode de musicien
        // . . . etc

        System.out.println("a et b référencent le même objet : "+(a==b));
        System.out.println("a référence l'adresse mémoire : "+a);
        System.out.println("b référence l'adresse mémoire : "+b);
        // fermer le context
        context.close();
    }
}
```