



# Cours Spring

Spring à l'action



# Configurer une application

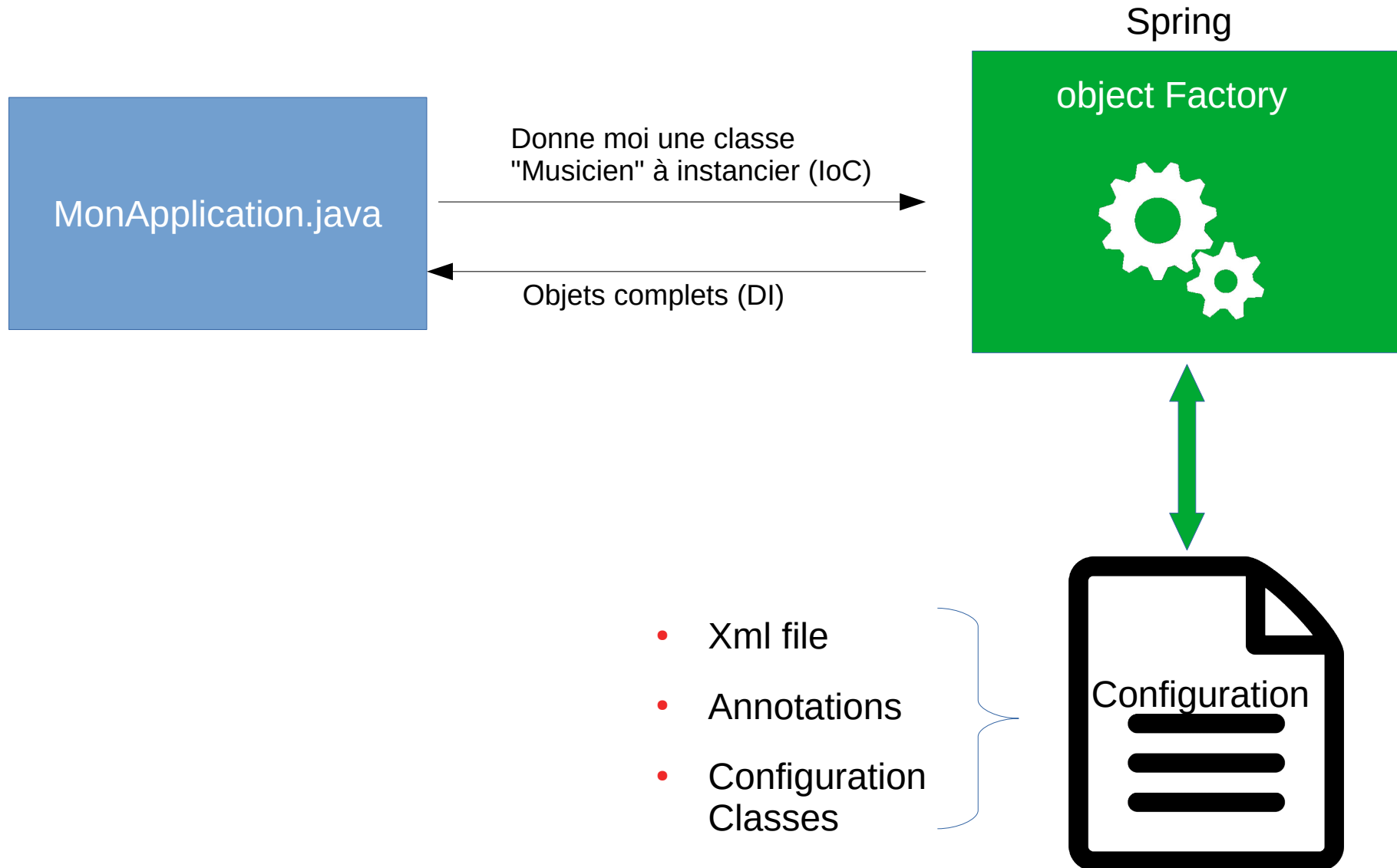
configurable != hard-coded

L'objectif c'est que l'application sache quelle est la classe à utiliser en lisant un fichier de configuration

Il suffira alors de modifier ce fichier de configuration pour instancier une classe différente (sans changer le code source).

Spring est conçu pour résoudre cette problématique.

# Spring Container



# XML configuration : Étape 1

Créer un fichier dans le classpath src/applicationContext.xml  
Y déclarer les objets à manipuler

'id' = une référence  
unique à la classe  
désignée

```
<?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">

  <bean id="monMusicien" class="com.springdemo.Pianiste" > </bean>

</beans>
```

Le chemin complet de  
la classe d'implémentation

## Etape 2. Créer le container Spring

- Vocabulaire : Spring Container = Contexte Spring
- On utilise une classe fournie par Spring pour créer le contexte : `ClassPathXmlApplicationContext`
- Cette classe possède un constructeur qui attend un paramètre : le nom du fichier xml de configuration

```
//load the spring configuration file
```

```
ClassPathXmlApplicationContext context = new  
ClassPathXmlApplicationContext("applicationContext.xml");
```



# Nb: création de contexte

Nb : chaque méthodes de configuration (xml, annotation ou classe de configuration ) possède une classe différente pour créer le contexte

- En xml (ici) c'est ClasspathXmlApplicationContext
- Avec les annotation c'est AnnotationConfig
- Et pour la configuration par classe java on utilisera WebApplicationContext

# Etape 3. récupérer les objets dans le code

Créer un container Spring

```
package com.springdemo;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MonApplication {

    public static void main(String[] args) {

        //load the spring configuration file
        ClassPathXmlApplicationContext context = new
        ClassPathXmlApplicationContext("applicationContext.xml");

        //retrieve a bean from the spring container
        Musicien musicien=context.getBean("monMusicien", Musicien.class);

        //call a method on the bean
        System.out.println(musicien.joueTaPartition());

        // close the application context
        context.close();

    }
}
```

"id" du bean

Interface

Accès au bean du container

Nb : le nom de classe instanciée n'apparaît plus dans le code (seulement dans le fichier de configuration XML)



# Executer l'application

- Rien n'a changé , tout fonctionne comme avant mais nous avons répondu à 100 % du besoin en atteignant le niveau de configuration souhaité... avec Spring.