

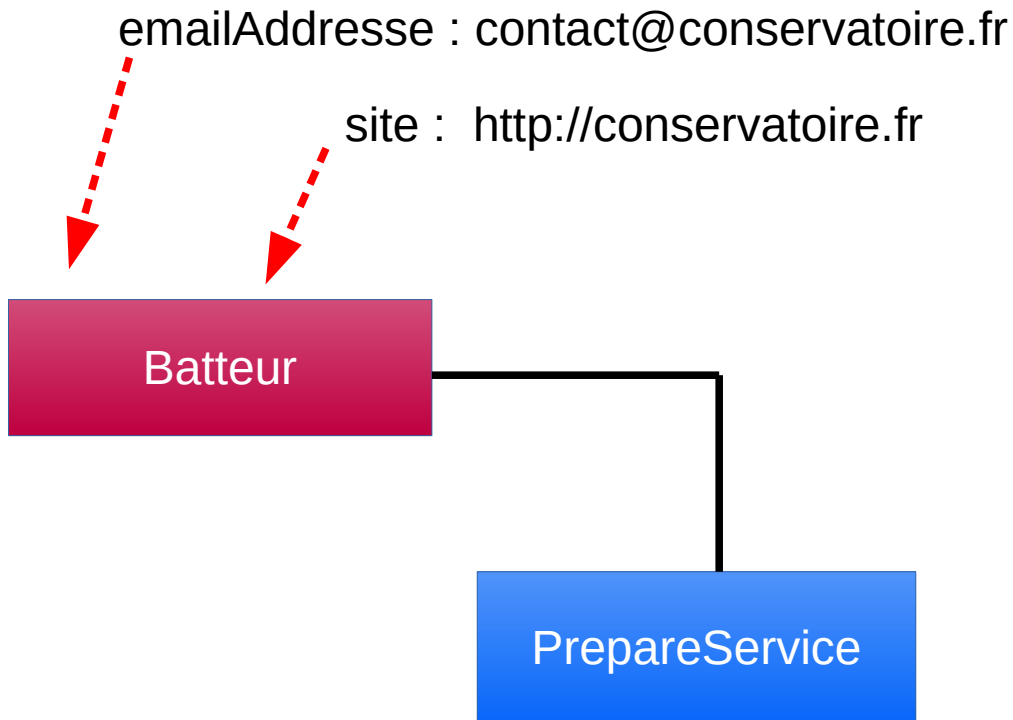


Cours Spring Core

2.3 Injection des valeurs littérales

Schéma

& Pas à pas



- Créer les champs, et les méthodes setter dans la classe d'injection
- Configurer ce comportement dans la configuration du contexte Spring

Créer le(s) propriété(s) et le(s) setter(s)

Configurer l'injection en xml

```
package com.springdemo;

public class Batteur implements Musicien {

    private String emailAdresse;
    private String site;

    ...

    public String getEmailAdresse() {
        return emailAdresse;
    }

    public void setEmailAdresse(String emailAdresse) {
        System.out.println("Batteur : à l'intérieur
de la méthode set - setEmailAdresse" );
        this.emailAdresse = emailAdresse;
    }

    public String getSite() {
        return site;
    }

    public void setSite(String site) {
        System.out.println("Batteur : à l'intérieur
de la méthode set - setSite");
        this.site = site;
    }

}
```

```
<?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">

    <!-- Définir les beans ici -->
    <!-- Définir les Dépendances -->
    <bean id="unPrepareService"
        class="com.springdemo.ZenPreparationService" >
    </bean>

    <bean id="unMusicien" class="com.springdemo.Violoniste" >

    <!-- Définir l' Injection par constructeur -->
    <constructor-arg ref="unPrepareService"/>
    </bean>

    <bean id="unBatteur" class="com.springdemo.Batteur" >
    <!-- Définir l'injection par setter -->
    <property name="prepareService" ref="unPrepareService" />

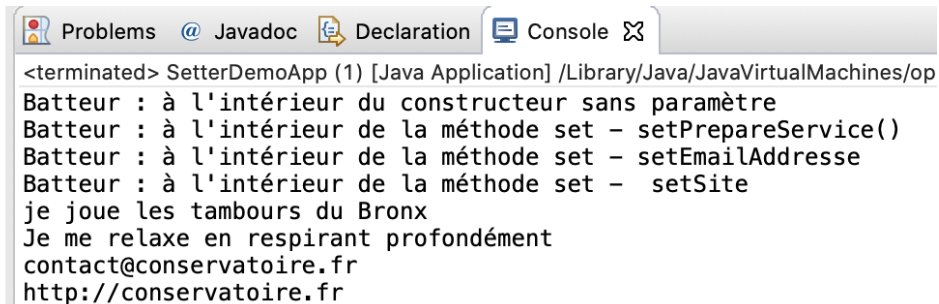
    <property name="emailAdresse" value="contact@conservatoire.fr" />
    <property name="site" value="http://conservatoire.fr" />
    </bean>

</beans>
```

Mise à jour de SetterDemoApp

```
public class SetterDemoApp {  
    public static void main(String[] args) {  
        // charger le contexte spring depuis le fichier de configuration xml  
        ClassPathXmlApplicationContext context = new  
ClassPathXmlApplicationContext("applicationContext.xml");  
  
        // récupérer un bean depuis le container  
        Batteur batteur = context.getBean("unBatteur", Batteur.class);  
  
        // appeler une méthode sur le bean  
        // ... on verra cela plus tard ...  
        System.out.println(batteur.joueTaPartition());  
        System.out.println(batteur.getPrepa());  
  
        // Appeler les variables dont les valeurs sont injectées en xml  
        System.out.println(batteur.getEmailAdresse());  
        System.out.println(batteur.getSite());  
  
        // fermer le contexte  
        context.close();  
    }  
}
```

Run As > Java Application =>



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of the Java application. The output shows the execution of the main method, including the creation of the Spring context, retrieval of the 'unBatteur' bean, and the execution of several methods on the bean. The output is as follows:

```
<terminated> SetterDemoApp (1) [Java Application] /Library/Java/JavaVirtualMachines/op  
Batteur : à l'intérieur du constructeur sans paramètre  
Batteur : à l'intérieur de la méthode set - setPrepareService()  
Batteur : à l'intérieur de la méthode set - setEmailAdresse  
Batteur : à l'intérieur de la méthode set - setSite  
je joue les tambours du Bronx  
Je me relaxe en respirant profondément  
contact@conservatoire.fr  
http://conservatoire.fr
```