

2021-2022

Département Mathématiques et Informatique  
GLSID 2 – S4

## Architecture JEE et Middlewares

### Rapport

TP 2 IOC en utilisant Spring

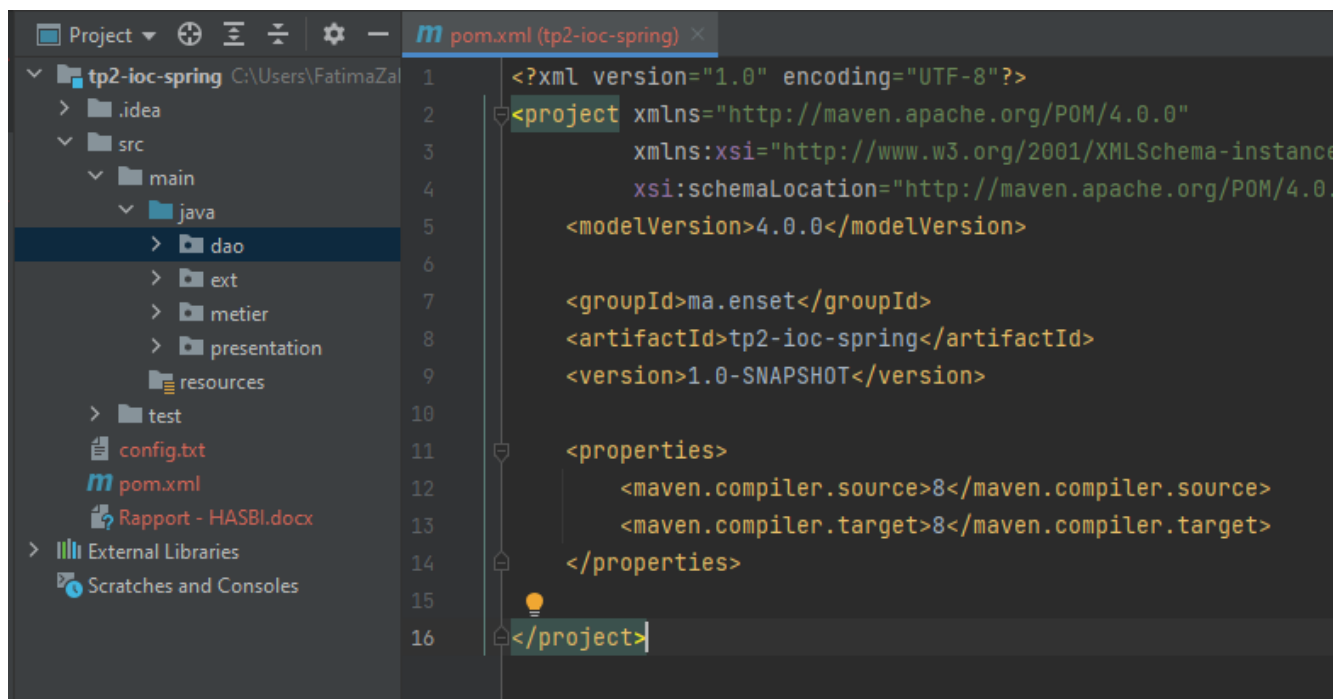
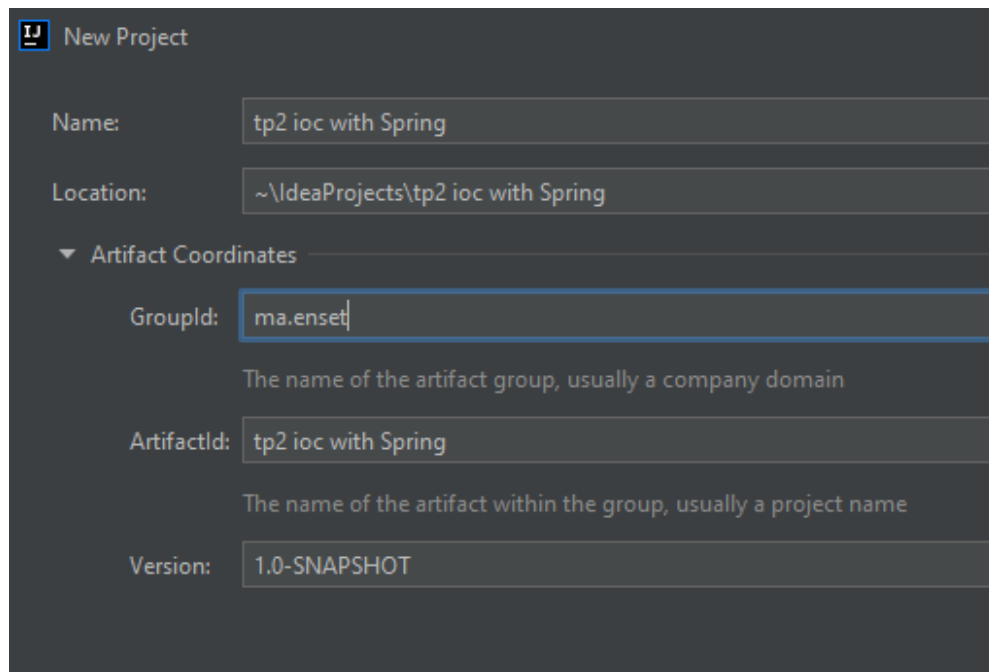
Version xml et version annotations

Préparé par : **Fatima Zahra HASBI**  
ENCADRE PAR : **MR MOHAMED**  
**YOUSSEFI**

Ce tp consiste à utiliser dans l'exercice 1 l'inversion de contrôle avec Spring (Xml version).

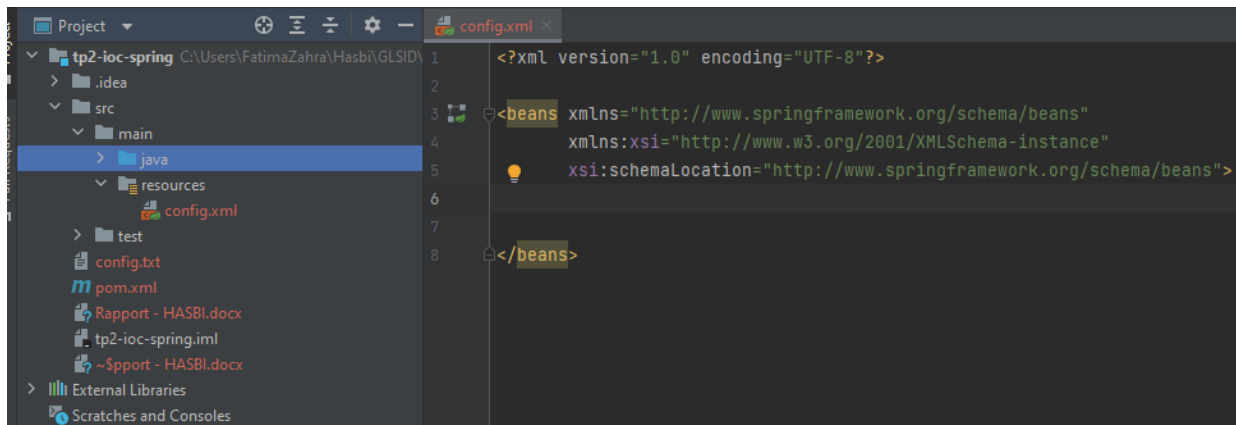
Dans ce rapport je montre un ensemble des captures d'écran de réalisation du tp.

## Création du projet Maven sur IntelliJ IDE

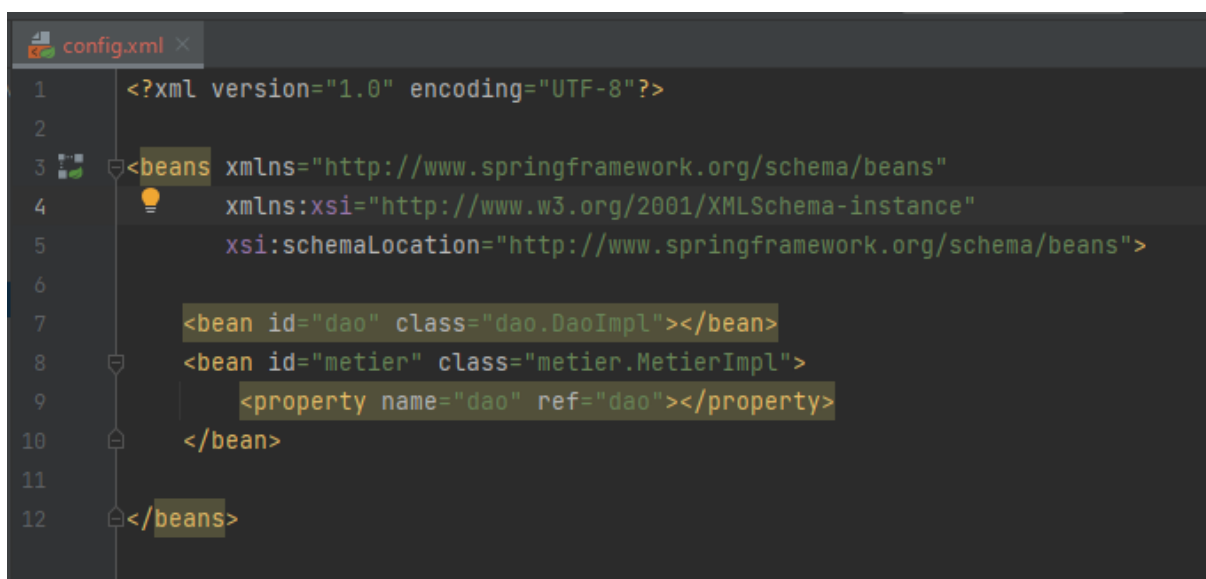


## Version xml :

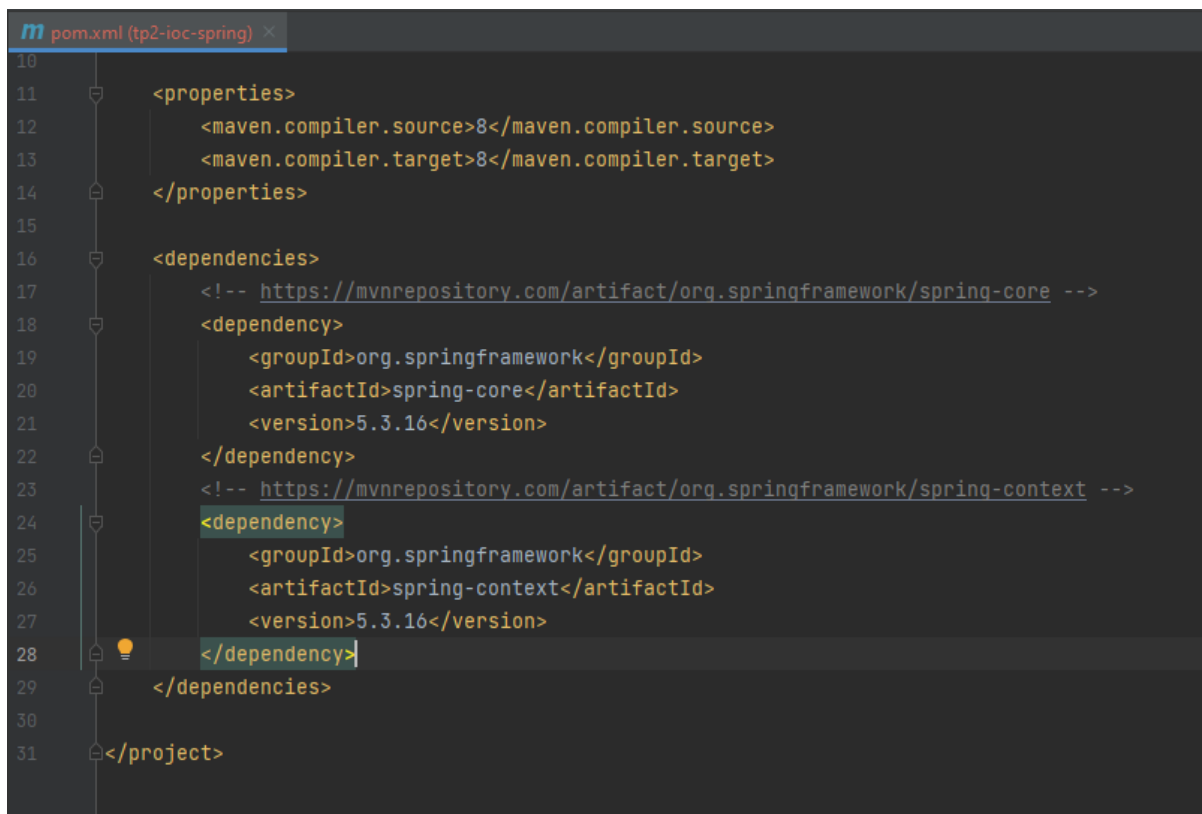
### Création d'un fichier de configuration Spring dans le dossier Ressource



### Les classes qu'on veut instancier pendant le démarrage de l'application

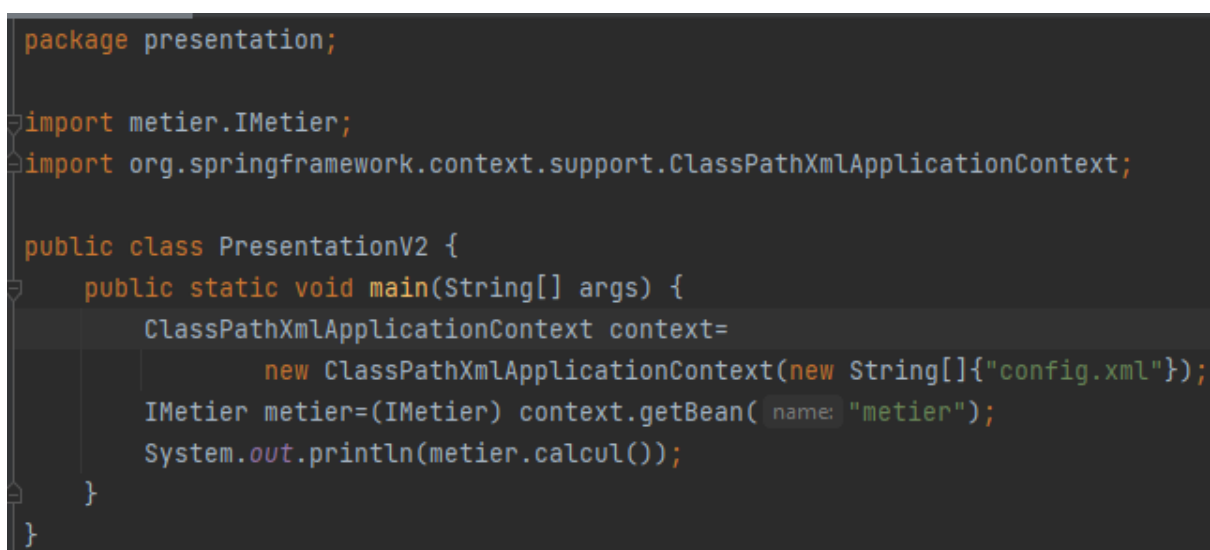


## Ajouter les dépendances de Spring dans le fichier pom.xml



```
m pom.xml (tp2-ioc-spring) x
10
11 <properties>
12     <maven.compiler.source>8</maven.compiler.source>
13     <maven.compiler.target>8</maven.compiler.target>
14 </properties>
15
16 <dependencies>
17     <!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
18     <dependency>
19         <groupId>org.springframework</groupId>
20         <artifactId>spring-core</artifactId>
21         <version>5.3.16</version>
22     </dependency>
23     <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
24     <dependency>
25         <groupId>org.springframework</groupId>
26         <artifactId>spring-context</artifactId>
27         <version>5.3.16</version>
28     </dependency>
29 </dependencies>
30
31 </project>
```

## Création de l'application dans la couche presentation

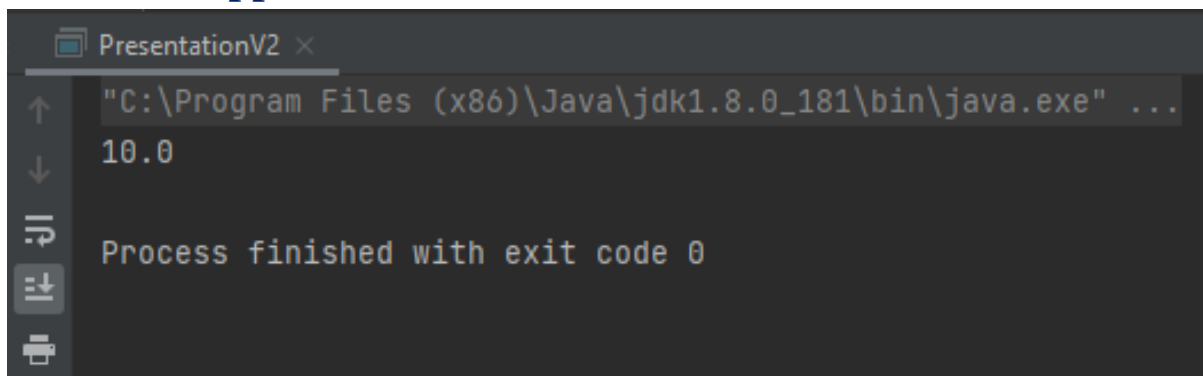


```
package presentation;

import metier.IMetier;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class PresentationV2 {
    public static void main(String[] args) {
        ClassPathXmlApplicationContext context=
            new ClassPathXmlApplicationContext(new String[]{"config.xml"});
        IMetier metier=(IMetier) context.getBean( name: "metier");
        System.out.println(metier.calcul());
    }
}
```

## Exécuter l'application



```
PresentationV2 x
"C:\Program Files (x86)\Java\jdk1.8.0_181\bin\java.exe" ...
10.0
Process finished with exit code 0
```

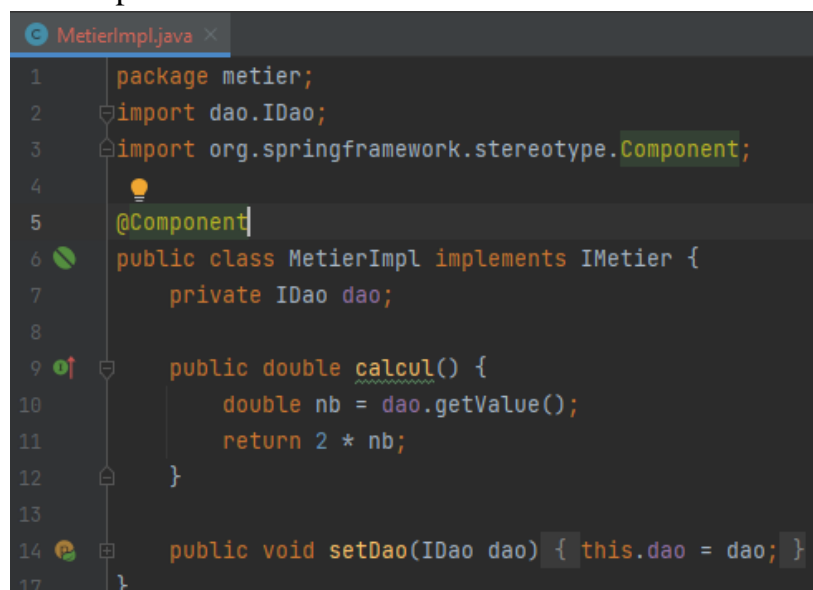
## Version annotations :

On ajoute l'annotation `@Component` pour la classe `IDaoImpl`



```
DaoImpl.java x
1 package dao;
2
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class DaoImpl implements IDao {
7     @Override
8     public double getValue() {
9         return 5;
10    }
```

Aussi pour `IMetierImpl`



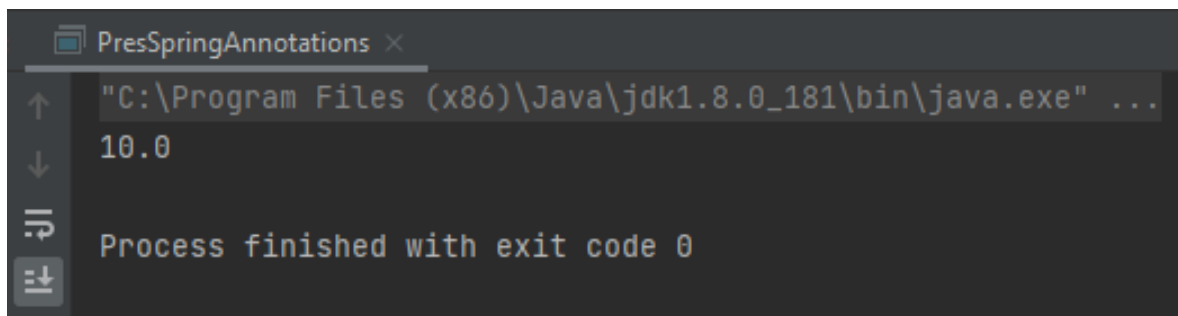
```
MetierImpl.java x
1 package metier;
2 import dao.IDao;
3 import org.springframework.stereotype.Component;
4
5 @Component
6 public class MetierImpl implements IMetier {
7     private IDao dao;
8
9     public double calcul() {
10         double nb = dao.getValue();
11         return 2 * nb;
12     }
13
14     public void setDao(IDao dao) { this.dao = dao; }
15 }
```

## Pour faire l'injection de dépendances on ajoute l'annotation @Autowired

```
MetierImpl.java x
1 package metier;
2 import dao.IDao;
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Component;
5
6 @Component
7 public class MetierImpl implements IMetier {
8
9     @Autowired
10    private IDao dao;
11
12    public double calcul() {
13        double nb = dao.getValue();
14        return 2 * nb;
15    }
16
17    public void setDao(IDao dao) { this.dao = dao; }
18
19 }
20
```

On va créer une nouvelle application dans la couche presentation pour travailler avec ces annotations

```
PresSpringAnnotations.java x
1 package presentation;
2
3 import metier.IMetier;
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
6
7 public class PresSpringAnnotations {
8
9     public static void main(String[] args) {
10        ApplicationContext ctx= new AnnotationConfigApplicationContext( ...basePackages: "dao", "meti
11        IMetier metier=ctx.getBean(IMetier.class);
12        System.out.println(metier.calcul());
13    }
14 }
```

**Le résultat de l'exécution :**

The screenshot shows a console window titled "PresSpringAnnotations" with a close button. The console output displays the Java command path and version: "C:\Program Files (x86)\Java\jdk1.8.0\_181\bin\java.exe" ... 10.0. Below this, it states "Process finished with exit code 0". On the left side of the console, there are four icons: an upward arrow, a downward arrow, a refresh/clear icon, and a search icon.

```
"C:\Program Files (x86)\Java\jdk1.8.0_181\bin\java.exe" ...  
10.0  
  
Process finished with exit code 0
```

**FIN**