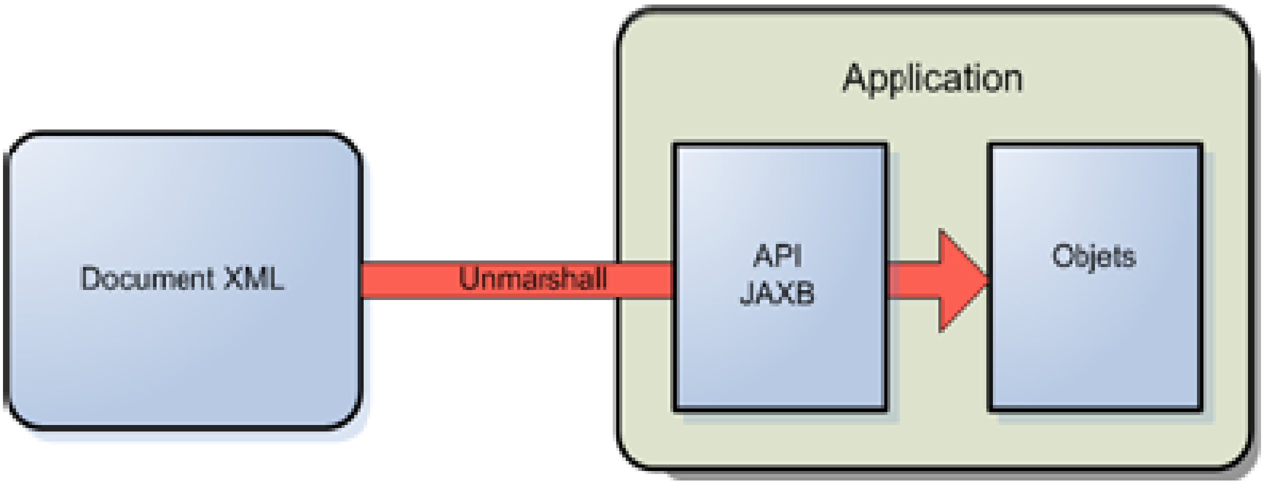
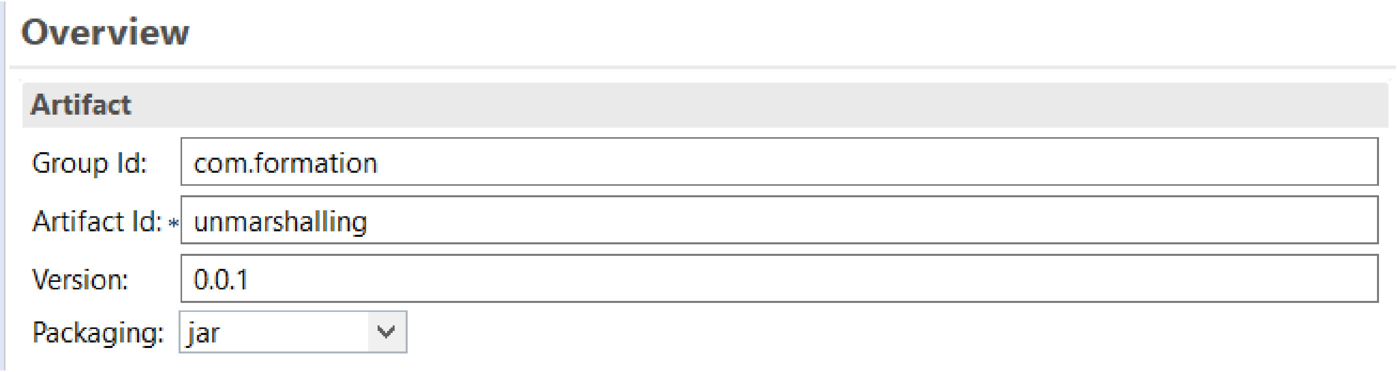
UnMarshalling



Créer un projet maven



Overview

Artifact

Group Id: com.formation

Artifact Id: * unmarshalling

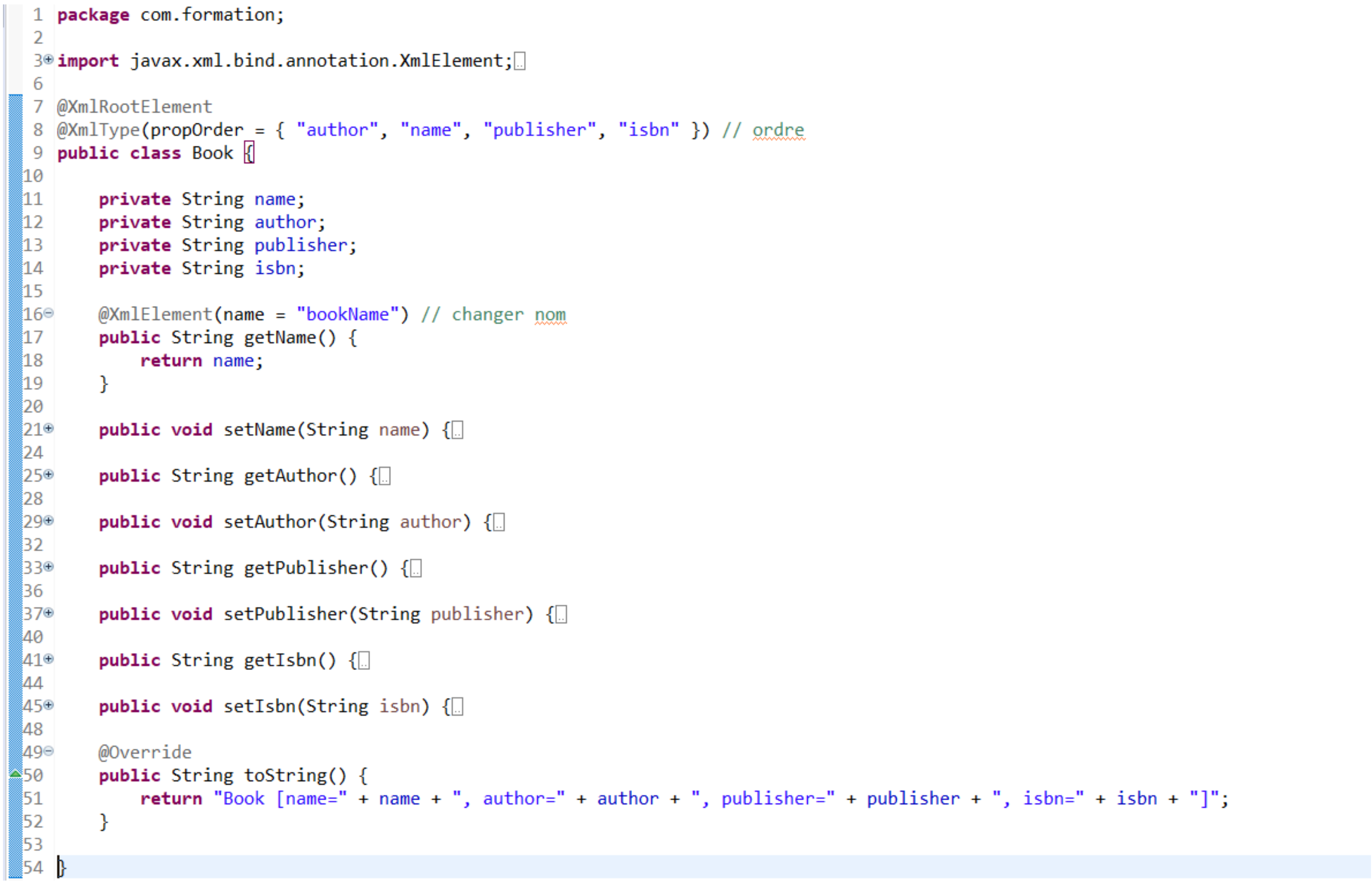
Version: 0.0.1

Packaging: jar

Ajouter dans le pom.xml

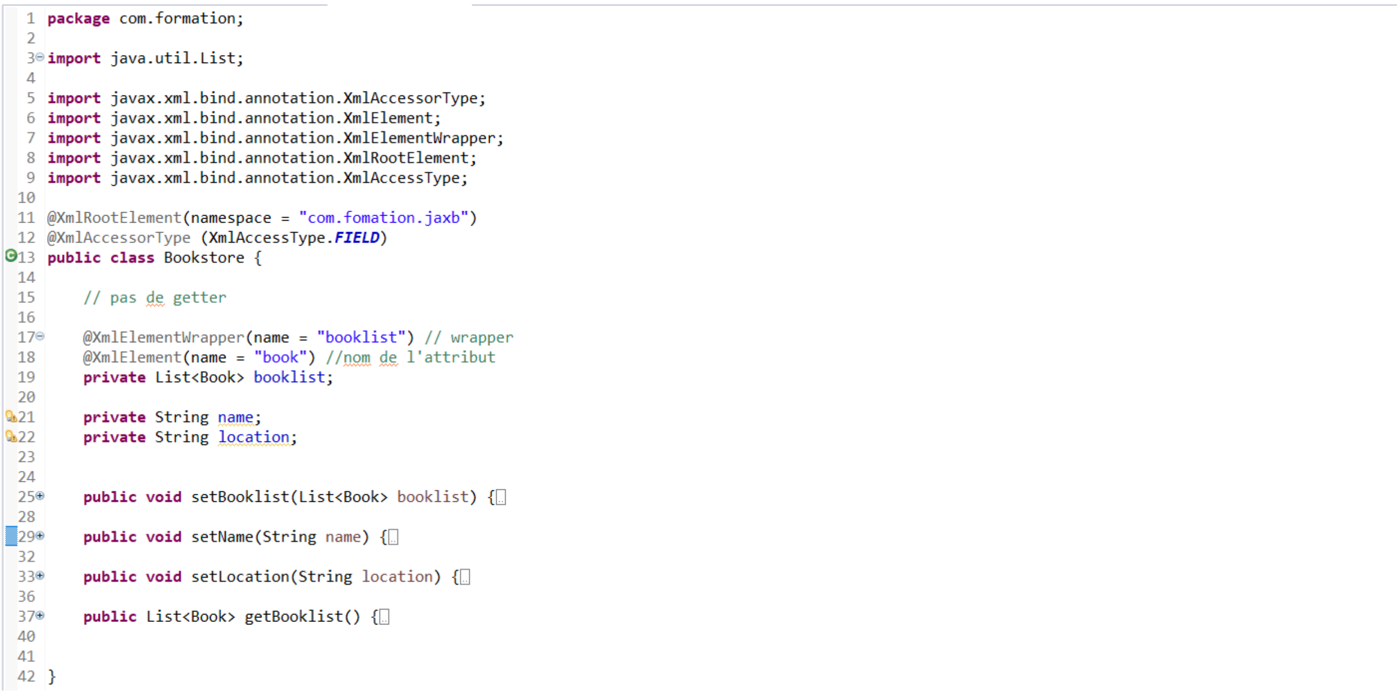
```
<properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
</properties>

<dependencies>
        <dependency>
                <groupId>javax.xml.bind</groupId>
                <artifactId>jaxb-api</artifactId>
                <version>2.2.11</version>
        </dependency>
        <dependency>
                <groupId>com.sun.xml.bind</groupId>
                <artifactId>jaxb-core</artifactId>
                <version>2.2.11</version>
        </dependency>
        <dependency>
                <groupId>com.sun.xml.bind</groupId>
                <artifactId>jaxb-impl</artifactId>
                <version>2.2.11</version>
        </dependency>
        <dependency>
                <groupId>javax.activation</groupId>
                <artifactId>activation</artifactId>
                <version>1.1.1</version>
        </dependency>

</dependencies>
```

Créer la classe Book



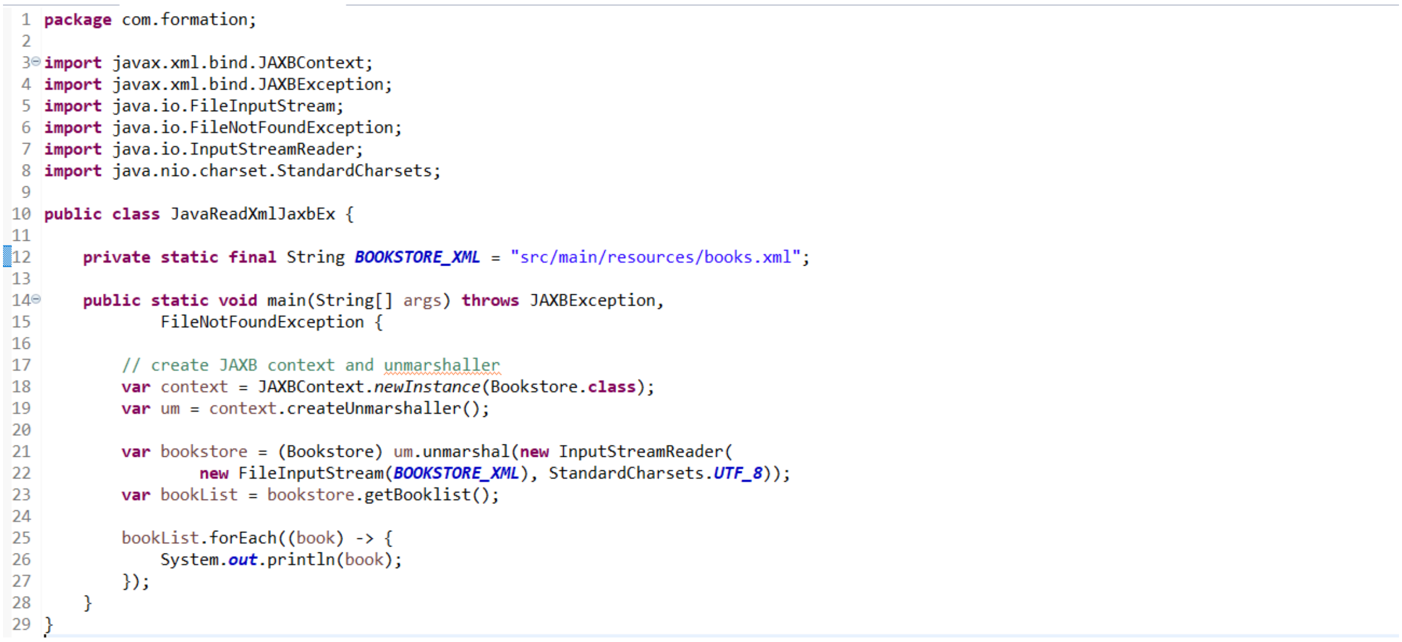
```java
package com.formation;

import javax.xml.bind.annotation.XmlElement;□

@XmlRootElement
@XmlType(propOrder = { "author", "name", "publisher", "isbn" }) // ordre
public class Book {

    private String name;
    private String author;
    private String publisher;
    private String isbn;

    @XmlElement(name = "bookName") // changer nom
    public String getName() {
        return name;
    }

    public void setName(String name) {□

    public String getAuthor() {□

    public void setAuthor(String author) {□

    public String getPublisher() {□

    public void setPublisher(String publisher) {□

    public String getIsbn() {□

    public void setIsbn(String isbn) {□

    @Override
    public String toString() {
        return "Book [name=" + name + ", author=" + author + ", publisher=" + publisher + ", isbn=" + isbn + "]";
    }

}
```

Créer la classe BookStore



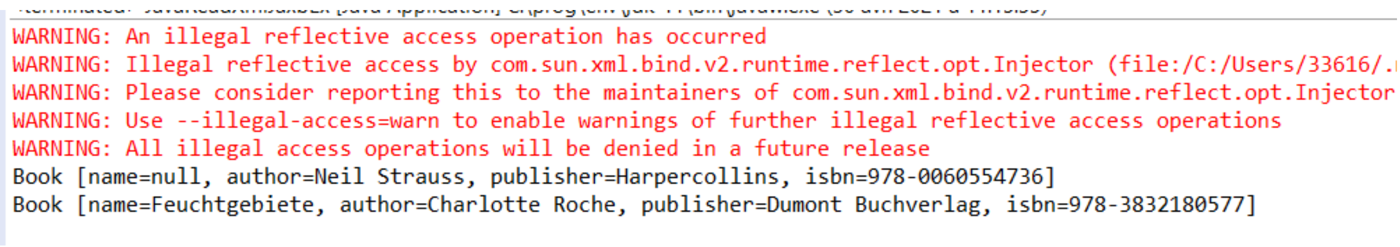
```java
package com.formation;

import java.util.List;

import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlElementWrapper;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlAccessType;

@XmlRootElement(namespace = "com.fomation.jaxb")
@XmlAccessorType (XmlAccessType.FIELD)
public class Bookstore {

    // pas de getter

    @XmlElementWrapper(name = "booklist") // wrapper
    @XmlElement(name = "book") //nom de l'attribut
    private List<Book> booklist;

    private String name;
    private String location;


    public void setBooklist(List<Book> booklist) {□

    public void setName(String name) {□

    public void setLocation(String location) {□

    public List<Book> getBooklist() {□


}
```

Tester



```java
package com.formation;

import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;

public class JavaReadXmlJaxbEx {

    private static final String BOOKSTORE_XML = "src/main/resources/books.xml";

    public static void main(String[] args) throws JAXBException,
            FileNotFoundException {

        // create JAXB context and unmarshaller
        var context = JAXBContext.newInstance(Bookstore.class);
        var um = context.createUnmarshaller();

        var bookstore = (Bookstore) um.unmarshal(new InputStreamReader(
                new FileInputStream(BOOKSTORE_XML), StandardCharsets.UTF_8));
        var bookList = bookstore.getBooklist();

        bookList.forEach((book) -> {
            System.out.println(book);
        });
    }
}
```

Nous obtenons



```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.sun.xml.bind.v2.runtime.reflect.opt.Injector (file:/C:/Users/33616/.
WARNING: Please consider reporting this to the maintainers of com.sun.xml.bind.v2.runtime.reflect.opt.Injector
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Book [name=null, author=Neil Strauss, publisher=Harpercollins, isbn=978-0060554736]
Book [name=Feuchtgebiete, author=Charlotte Roche, publisher=Dumont Buchverlag, isbn=978-3832180577]
```