**Marshalling/ Un-marshalling XML to POJO**

1. **JAXB:**

JAXB stands for Java Architecture for XML Binding. It provides a mechanism to write Java objects into XML and read XML as objects. Simply put, you can say it is used to convert Java objects into XML and vice-versa.

JAXB provides a fast and convenient way to bind XML schemas and Java representations, making it easy for Java developers to incorporate XML data and processing functions in Java applications. As part of this process, JAXB provides methods for unmarshalling (reading) XML instance documents into Java content, and then marshalling (writing) Java content back into XML instance documents. JAXB also provides a way to generate XML schemas from Java objects.

1. **Features of JAXB 2.0:**

* Support for all W3C XML Schema features.
* Annotation support with the addition of the javax.xml.bind.annotation package to control this binding.
* Additional validation capabilities through the JAXP 1.3 validation APIs.
* Smaller runtime libraries.

1. **Important annotations:**

* @XmlRootElement specifies the root element for the XML document.
* @XmlAttribute specifies the attribute for the root element.
* @XmlElement specifies the sub-element for the root element.

1. **Code for Marshalling:**

javax.xml.bind.annotation.\*;

public void marshall(){

try{

Employee emp = new Employee("A001", "Manoj", "Nashik", 50000);

//creating the JAXB context

JAXBContext jc = new JAXBContext.newInstance(Employee.class);

//creating the unmarshall object

Marshaller ms =jc.createMarshaller();

//setting the property to show xml format output

ms.setProperty(Marshaller.JAXB\_FORMATTED\_OUTPUT, true);

//calling the marshall method

ms.marshal(emp, System.out);

ms.marshal(emp, new File("src\\data\\Employee.xml"));

}

catch(Exception e){

System.out.println(" "+e.getMessage());

}

}

1. **Unmarshalling:**

public void unmarshall(){

try{

//getting the xml file to read

File file = new File(“/home/knoldus/Desktop/student.xml”);

//creating the JAXB context

JAXBContext jContext = JAXBContext.newInstance(Student.class);

//creating the unmarshall object

Unmarshaller unmarshallerObj = jContext.createUnmarshaller();

//calling the unmarshall method

Student student=(Student) unmarshallerObj.unmarshal(file);

System.out.println(student.getName()+” “+student.getId()+” “+student.getSubject());

}catch(Exception e){

e.printStackTrace();

}

}