JAXB: MARSHALLING AND UNMARSHALLING EXAMPLE

10 October 2012 By [Nithya Vasudevan](https://plus.google.com/110690891983832851434?rel=author) 14,869 views 4 Comments

**10**FLARES Twitter 1[Tweet](https://twitter.com/share)Facebook 2Google+7

**Contents**[[hide](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/)]

* [1 Project Description](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Project_Description)
* [2 Environment Used:](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Environment_Used)
* [3 New project in Eclipse](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#New_project_in_Eclipse)
  + [3.1 Create Bean classes](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Create_Bean_classes)
    - [3.1.1 Address.java](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Addressjava)
    - [3.1.2 Employee.java](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Employeejava)
  + [3.2 Create JAXB Handler (Helper class)](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Create_JAXB_Handler_Helper_class)
  + [3.3 Create Java Application client (main())](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Create_Java_Application_client_main)
* [4 Output](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Output)
* [5 Folder Structure](http://theopentutorials.com/tutorials/java/jaxb/jaxb-marshalling-and-unmarshalling-example/#Folder_Structure)

Project Description

* In this JAXB tutorial we will see an example on how to marshal and unmarshal Java objects.
* This example explains how to,
  + map Java object’s property to XML element’s attribute using @XmlAttribute annotation,
  + map Java object’s property that references a class annotated with @XmlType as a child element to XML parent element.

Environment Used:

* JDK 6.
* Eclipse Indigo IDE for Java EE Developers.

New project in Eclipse

Create a new Java project in Eclipse IDE and name it as **“SimpleJAXBDemo”**.

**Create Bean classes**

Address.java

Create a new Class “**Address.java**” in the package “**com.theopentutorials.jaxb.to**” and copy the following code.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | package com.theopentutorials.jaxb.to;    import javax.xml.bind.annotation.XmlType;    @XmlType  public class Address {      private String street;      private String city;      private String state;      private long zip;      public String getStreet() {          return street;      }      public void setStreet(String street) {          this.street = street;      }      public String getCity() {          return city;      }      public void setCity(String city) {          this.city = city;      }      public String getState() {          return state;      }      public void setState(String state) {          this.state = state;      }      public long getZip() {          return zip;      }      public void setZip(long zip) {          this.zip = zip;      }      @Override      public String toString() {          return "Address [street=" + street + ", city=" + city + ", state="                  + state + ", zip=" + zip + "]";      }  } |

This is a bean class which is embedded in parent class (Employee) to create a child element (<address> tag). This class is annotated with @javax.xml.bind.annotation.XmlType.

Employee.java

Create a new Class “**Employee.java**” in the package “**com.theopentutorials.jaxb.to**” and copy the following code.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57 | package com.theopentutorials.jaxb.to;    import javax.xml.bind.annotation.XmlAccessType;  import javax.xml.bind.annotation.XmlAccessorType;  import javax.xml.bind.annotation.XmlAttribute;  import javax.xml.bind.annotation.XmlRootElement;    @XmlAccessorType(XmlAccessType.FIELD)  @XmlRootElement(name = "employee")  public class Employee {        @XmlAttribute      private int id;      private String name;      private String department;      private Address address;        //Must have no-argument constructor      public Employee() { }        public Employee(int id, String name, String department, Address address) {          super();          this.id = id;          this.name = name;          this.department = department;          this.address = address;      }      public int getId() {          return id;      }      public void setId(int id) {          this.id = id;      }      public String getName() {          return name;      }      public void setName(String name) {          this.name = name;      }      public String getDepartment() {          return department;      }      public void setDepartment(String department) {          this.department = department;      }      public Address getAddress() {          return address;      }      public void setAddress(Address address) {          this.address = address;      }      @Override      public String toString() {          return "Employee [id=" + id + ", name=" + name + ", department="                  + department + ", address=" + address + "]";      }  } |

This is the root/parent class containing JAXB annotations whose object is marshalled and unmarshalled.

* **@XmlAccessorType** specifies whether annotations are applied to fields or Javabean properties (getters and setters).
* **@XmlRootElement** specifies the XML element.
* **@XmlAttribute** specifies XML element’s attribute.

**Create JAXB Handler (Helper class)**

Create a new Class “**JAXBXMLHandler.java**” in the package “**com.theopentutorials.jaxb.xml**” and copy the following code.

This is a helper class which has methods to perform marshalling and unmarshalling. These methods are called from client code (in this case, main() method).

* **Marshalling** is the process of writing Java objects to XML file.
* **Unmarshalling** is the process of converting XML content to Java objects.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | package com.theopentutorials.jaxb.xml;    import java.io.BufferedWriter;  import java.io.File;  import java.io.FileWriter;  import java.io.IOException;  import javax.xml.bind.JAXBContext;  import javax.xml.bind.JAXBException;  import javax.xml.bind.Marshaller;  import javax.xml.bind.Unmarshaller;  import com.theopentutorials.jaxb.to.Employee;    public class JAXBXMLHandler {        // Export: Marshalling      public static void marshal(Employee employee, File selectedFile)              throws IOException, JAXBException {          JAXBContext context;          BufferedWriter writer = null;          writer = new BufferedWriter(new FileWriter(selectedFile));          context = JAXBContext.newInstance(Employee.class);          Marshaller m = context.createMarshaller();          m.setProperty(Marshaller.JAXB\_FORMATTED\_OUTPUT, true);          m.marshal(employee, writer);          writer.close();      }        // Import: Unmarshalling      public static Employee unmarshal(File importFile) throws JAXBException {          Employee employee = null;          JAXBContext context;            context = JAXBContext.newInstance(Employee.class);          Unmarshaller um = context.createUnmarshaller();          employee = (Employee) um.unmarshal(importFile);            return employee;      }  } |

**Create Java Application client (main())**

Create a new Class “**JAXBDemo.java**” in the package “**com.theopentutorials.jaxb.main**” and copy the following code.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package com.theopentutorials.jaxb.main;    import java.io.File;  import java.io.IOException;  import javax.xml.bind.JAXBException;  import com.theopentutorials.jaxb.to.Address;  import com.theopentutorials.jaxb.to.Employee;  import com.theopentutorials.jaxb.xml.JAXBXMLHandler;    public class JAXBDemo {      public static void main(String[] args) {            Address address = new Address();          address.setStreet("2163, 1st Avenue");          address.setCity("Peoria");          address.setState("Illinois");          address.setZip(61606);            Employee employee = new Employee(1, "Kumar", "Development", address);            try {              //Marshalling: Writing Java object to XML file              JAXBXMLHandler.marshal(employee, new File("employee.xml"));                //Unmarshalling: Converting XML content to Java objects              Employee employee2 = JAXBXMLHandler.unmarshal(new File("employee.xml"));              System.out.println(employee2);          } catch (IOException e) {              e.printStackTrace();          } catch (JAXBException e) {              e.printStackTrace();          }      }  } |

* This class creates employee object and calls marshal method from JAXBXMLHandler helper class passing the employee object and the file to write the generated xml.
* We call unmarshal method passing the marshalled file name which returns employee object.
* Finally, printing the employee object (which calls toString() from Employee class).

Output

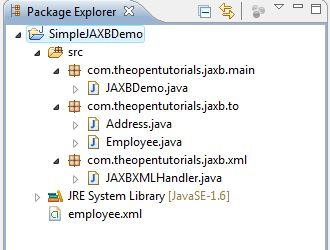
Run the JAXBDemo.java.

Employee [id=1, name=Kumar, department=Development,  
address=Address [street=2163, 1st Avenue, city=Peoria, state=Illinois, zip=61606]]

Refresh your project in Project Explorer (press F5 on your project) to see the generated XML file.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  <employee id="1">      <name>Kumar</name>      <department>Development</department>      <address>          <city>Peoria</city>          <state>Illinois</state>          <street>2163, 1st Avenue</street>          <zip>61606</zip>      </address>  </employee> |

Folder Structure

The complete folder structure of this example is shown below.  
[](http://theopentutorials.com/totwp331/wp-content/uploads/jaxb-marshalling-and-unmarshalling-example_3314/jaxb-marshal-unmarshall-eclipse-folder-structure.jpg?c3a9c1)