Representaciones XML y serialización de objetos (JAXB)

Contenido

[JAXB y anotaciones básicas 1](#_Toc147213141)

[Una empresa con un departamentos y este con varios empleados 2](#_Toc147213142)

[Ejercicio. Implementa Una empresa con varios departamentos y este con varios empleados 5](#_Toc147213143)

[Dependencias 8](#_Toc147213144)

# JAXB y anotaciones básicas

JAXB es una tecnología Java que permite mapear clases Java a representaciones XML y viceversa.

Al proceso de serializar un objeto Java a XML se llama **marshall o marshalling**

Al proceso de deserializar XML a un objeto Java se llama **unmarshall o unmarshalling**

Dos formas de mapear las clases:

1. Mapear clases java (beans) a representaciones XML con ANOTACIONES (@anotación)
2. Mapear clases Java a representaciones XML con la ayuda de un archivo SCHEMA XML (.xsd)

Nosotros lo haremos por anotaciones. Algunas anotaciones:

* **@XmlRootElement:** Esta anotación se utiliza en una clase Java para indicar que la clase representa el elemento raíz de un documento XML. Cada documento XML debe tener un elemento raíz, y esta anotación se utiliza para especificar qué clase Java se utilizará para representar dicho elemento.   
  @XmlRootElement

public class Empresa { …

* **@XmlElement:** Esta anotación se usa en campos o métodos getter de una clase Java para especificar cómo se debe mapear un elemento XML a esa propiedad. Puedes utilizar esta anotación para personalizar el nombre del elemento XML, el orden, y más.

**@XmlElement** (name="nombre"); --> permite cambiar el nombre de los elementos cuando el fichero XML se construya.

Ejemplo:

@XmlElement

public String getNombre() { …

@XmlElement(name=”nombreDept”)

public String getNombre() {

return nombre;

}

* **@XmlAttribute:** Esta anotación se utiliza en campos o métodos getter para indicar que un campo debe ser mapeado como un atributo XML en lugar de un elemento. Puedes personalizar el nombre del atributo XML y más utilizando esta anotación.
* **@XmlType**: Se utiliza en una clase Java para personalizar la información de tipo XML, como el nombre del tipo XML, el orden de los elementos y más. . Se anota en la clase que no va a ser raíz en el XML.

**@XmlType(propOrder = {campo1, campo2, ....)** 🡪 define los elementos del documento y en qué orden se van a presentar

NOTA. Esta anotación puede omitirse si no se quiere modificar y por tanto, el nombre de los elementos en el XML coincidirá con el nombre de los atributos de la clase.

* Si un atributo es una colección (List, ArrayList ...) debe llevar dos anotaciones:

**@XmlElementWrapper(name="elemento\_ envolvente")   
@XmlElement (name="elemento")**

* **Otras anotaciones:**
  + **@XmlAccessorType:** Esta anotación se utiliza en una clase o en un paquete para especificar cómo se deben tratar los campos y propiedades de la clase al mapearlos a XML. Puedes configurarlo para que JAXB utilice campos, propiedades o ambos.
  + **@XmlSeeAlso:** Se usa en una clase para especificar otras clases relacionadas que deben incluirse en el contexto JAXB. Esto es útil cuando tienes clases heredadas o interfaces que deben ser consideradas durante el proceso de mapeo.
  + **@XmlTransient**: Se aplica a campos o métodos getter para indicar que un campo o propiedad no debe mapearse a XML y debe ser ignorado durante el proceso de mapeo.

# Una empresa con un departamentos y este con varios empleados

package modelo;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement(name = "empresa")

**public class Empresa {**

private String nombre;

private Departamento departamento;

public Empresa() {

}

public Empresa(String nombre, Departamento departamento) {

this.nombre = nombre;

this.departamento = departamento;

}

public String getNombre() {

return nombre;

}

public Departamento getDepartamento() {

return departamento;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public void setDepartamento(Departamento departamento) {

this.departamento = departamento;

}

}

package modelo;

import java.util.ArrayList;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlElementWrapper;

import javax.xml.bind.annotation.XmlType;

@XmlType(propOrder = {"numDep", "nombre", "localidad", "empleados"})

**public class Departamento {**

private int numDep;

private String nombre;

private String localidad;

private ArrayList<Empleado> empleados;

public void setEmpleados(ArrayList<Empleado> empleados) {

this.empleados = empleados;

}

public Departamento(int numDep, String nombre, String localidad, ArrayList<Empleado> empleados) {

this.numDep = numDep;

this.nombre = nombre;

this.localidad = localidad;

this.empleados = empleados;

}

public Departamento() {

}

public Departamento(int numDep, String nombre, String localidad) {

this.numDep = numDep;

this.nombre = nombre;

this.localidad = localidad;

}

public void setNumDep(int numDep) {

this.numDep = numDep;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public void setLocalidad(String localidad) {

this.localidad = localidad;

}

public int getNumDep() {

return numDep;

}

public String getNombre() {

return nombre;

}

public String getLocalidad() {

return localidad;

}

@XmlElementWrapper(name = "empleados")

@XmlElement(name = "emple")

public ArrayList<Empleado> getEmpleados() {

return empleados;

}

}

package modelo;

import javax.xml.bind.annotation.XmlType;

@XmlType(propOrder = {"numEmple","nombre", "oficio", "salario"})

**public class Empleado {**

private int numEmple;

private String nombre;

private String oficio;

private long salario;

public Empleado() {

}

public Empleado(int numEmple, String nombre, String oficio, long salario) {

this.numEmple = numEmple;

this.nombre = nombre;

this.oficio = oficio;

this.salario = salario;

}

public int getNumEmple() {

return numEmple;

}

public String getNombre() {

return nombre;

}

public String getOficio() {

return oficio;

}

public long getSalario() {

return salario;

}

public void setNumEmple(int numEmple) {

this.numEmple = numEmple;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public void setOficio(String oficio) {

this.oficio = oficio;

}

public void setSalario(long salario) {

this.salario = salario;

}

}

package principal;

import java.io.File;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Unmarshaller;

import modelo.Empleado;

import modelo.Empresa;

**public class Inicio {**

public static void main(String[] args) {

Empresa empresa;

try {

//Sube el contenido del archivo xml al objeto empresa para su pocesarmiento

JAXBContext jc = JAXBContext.newInstance(Empresa.class); //clase raiz

Unmarshaller um = jc.createUnmarshaller();

empresa = (Empresa) um.unmarshal(new File("./ficheros/empleydepar.xml"));

// se procesa el contenido

System.out.println(empresa.getNombre() + " -- " + empresa.getDepartamento().getLocalidad());

for (Empleado e : empresa.getDepartamento().getEmpleados()) {

System.out.println(e.getNombre() + " -- " + e.getSalario());

}

} catch (JAXBException ex) {

Logger.getLogger(Inicio.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

# Ejercicio. Implementa Una empresa con varios departamentos y este con varios empleados

**Solución**

package es.bosco.probandojaxb;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

import java.util.List;

import javax.xml.bind.annotation.XmlElementWrapper;

@XmlRootElement(name = "empresa")

**public class Empresa {**

private String nombre;

private List<Departamento> departamentos;

@XmlElement

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

@XmlElementWrapper(name = "departamentos")

@XmlElement(name = "departamento")

public List<Departamento> getDepartamentos() {

return departamentos;

}

public void setDepartamentos(List<Departamento> departamentos) {

this.departamentos = departamentos;

}

}

package es.bosco.probandojaxb;

import javax.xml.bind.annotation.XmlElement;

import java.util.List;

import javax.xml.bind.annotation.XmlElementWrapper;

import javax.xml.bind.annotation.XmlType;

//@XmlType(propOrder = {"numDep", "nombre", "localidad", "empleados"})

**public class Departamento {**

private int numDep;

private String nombre;

private String localidad;

private double presupuesto;

private List<Empleado> empleados;

@XmlElement

public int getNumDep() {

return numDep;

}

public void setNumDep(int numDep) {

this.numDep = numDep;

}

@XmlElement

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

@XmlElement

public String getLocalidad() {

return localidad;

}

public void setLocalidad(String localidad) {

this.localidad = localidad;

}

@XmlElement

public double getPresupuesto() {

return presupuesto;

}

public void setPresupuesto(double presupuesto) {

this.presupuesto = presupuesto;

}

// @XmlElement(name = "emple")

@XmlElementWrapper(name = "empleados")

@XmlElement(name = "emple")

public List<Empleado> getEmpleados() {

return empleados;

}

public void setEmpleados(List<Empleado> empleados) {

this.empleados = empleados;

}

}

package es.bosco.probandojaxb;

import javax.xml.bind.annotation.XmlElement;

**public class Empleado {**

private int numEmple;

private String nombre;

private String oficio;

private double salario;

@XmlElement

public int getNumEmple() {

return numEmple;

}

public void setNumEmple(int numEmple) {

this.numEmple = numEmple;

}

@XmlElement

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

@XmlElement

public String getOficio() {

return oficio;

}

public void setOficio(String oficio) {

this.oficio = oficio;

}

@XmlElement

public double getSalario() {

return salario;

}

public void setSalario(double salario) {

this.salario = salario;

}

}

package es.bosco.probandojaxb;

import java.io.File;

import java.util.List;

import javax.xml.bind.JAXBContext;

import javax.xml.bind.JAXBException;

import javax.xml.bind.Unmarshaller;

**public class App {**

public static void main(String[] args) {

try {

// Crear un contexto JAXB para la clase Empresa

JAXBContext context = JAXBContext.newInstance(Empresa.class);

// Crear el objeto Unmarshaller para deserializar el archivo XML

Unmarshaller unmarshaller = context.createUnmarshaller();

// Cargar el archivo XML

File xmlFile = new File("./Empresa.xml");

// Deserializar el archivo XML en un objeto Empresa

Empresa empresa = (Empresa) unmarshaller.unmarshal(xmlFile);

System.out.println(empresa.getNombre());

// Obtener la lista de departamentos

List<Departamento> departamentos = empresa.getDepartamentos();

// Iterar sobre los departamentos e imprimir la información

for (Departamento departamento : departamentos) {

System.out.println("Nombre del departamento: " + departamento.getNombre());

// Obtener la lista de empleados

List<Empleado> empleados = departamento.getEmpleados();

// Imprimir el nombre de los empleados y contarlos

System.out.println("Empleados:");

for (Empleado empleado : empleados) {

System.out.println(" - " + empleado.getNombre());

}

System.out.println("Total de empleados: " + empleados.size());

System.out.println(); // Separador entre departamentos

}

} catch (JAXBException e) {

e.printStackTrace();

}

}

}

# Dependencias

<dependencies>

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>

<dependency>

<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>2.3.1</version>

</dependency>

</dependencies>