

Indicaciones para el manejo de Jaxb en NetBeans

El compilador JAXB permite generar una serie de clases Java, que podrán ser llamadas desde nuestras aplicaciones a través de los métodos sets y gets para obtener o establecer los datos de un documento XML.

El funcionamiento al usar JAXB sería:

- Crear un fichero .xsd (esquema) que contenga la estructura de las clases que vamos a utilizar.
- A partir del xsd se crean los POJOS, es decir una clase por cada tipo que hayamos especificado en el xsd. Después de creadas hay que compilarlas.
- Crear un fichero xml que responda a las especificaciones del xsd.
- Ahora ya se puede parsear el documento xml accediendo a los gets y sets generados en los POJOS. se podrán modificar o añadir datos.
- Tras realizar los cambios, se puede sobrescribir el documento xml.

Realización práctica:

- Debemos tener creado un xsd y xml válido para el xsd.
- Creamos un proyecto p.e. JAXB.albaran
- Creamos un enlace JAXB:
 - B.D. sobre el proyecto->nuevo->otro->XML->JAXB enlace
 - Damos nombre al enlace y seleccionamos el fichero xsd. Como nombre de paquete damos p.e. jaxb.albaran.
 - Al pinchar el botón Terminar se generan automáticamente las clases POJOS que habrá que compilar a continuación.
 - A continuación se crea la clase principal que habrá que compilar
 - Al ejecutar la clase principal se sobrescribe en el xml con las variaciones establecidas en la clase principal.

clase principal

```
public class ModificaAlbaPed {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        try {  
            //Crear una instancia para poder manipular las clases generadas, que están en el  
paquete jaxb.albaran  
            JAXBContext jaxbContext = JAXBContext.newInstance("jaxb.albaran");  
            //Crear un objeto de tipo Unmarshaller par convertir datos XML en un árbol de  
objetos Java.  
            Unmarshaller u = jaxbContext.createUnmarshaller();  
            //La clase JAXBElement representa un elemento de un documento xml, en este  
caso del documento albaran.xml  
            JAXBElement jaxbElement = (JAXBElement) u.unmarshal(  
                new FileInputStream("albaran.xml"));  
            //el método getValue() retorna el modelo contenido (content model) y el valor de los  
atributos del elemento  
            PedidoType pedidoType = (PedidoType) jaxbElement.getValue();
```

```

//Obtenemos una instancia del tipo PedidoType para obtener un objeto de tipo
Direccion
    Direccion direccion = pedidoType.getFacturarA();
//Establecemos los datos
    direccion.setNombre("Jose Antonio");
    direccion.setCalle("Zafiro 3");
    direccion.setCiudad("Molina");
    direccion.setProvincia("Murcia");
    direccion.setCodigoPostal(new BigDecimal("30500"));
//Crear un objeto de tipo Marshaller para posteriormente convertir el árbol de
objetos a datos XML.
    Marshaller m = jaxbContext.createMarshaller();
//Crear el resultado XML no formaeado para lectura de las personas, con saltos de
línea,etc
    m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE);
//Escribir el elemento obtenido como primer parámetro por la salida standard, el
segundo parametro
    m.marshal(jaxbElement, System.out);
} catch (JAXBException je) {
    System.out.println(je.getCause());
} catch (IOException ioe) {
    System.out.println(ioe.getMessage());
}
}
}
}

```