

Acceso a ficheros XML con SAX: lectura

- ▶ La lectura de un fichero XML produce una serie de eventos relacionados con el inicio y la finalización de cada una de las etiquetas, o con la aparición de caracteres entre las etiquetas, por ejemplo `startDocument()`, `endDocument()`, `startElement()`, `endElement()`, `characters()`.

- ▶ Estos eventos son tratados mediante la llamada a métodos.

- ▶ Se comienza creando una factoría para el procesador de XML.

```
SAXParserFactory parserFactory = SAXParserFactory.newInstance();
```

```
SAXParser parser = parserFactory.newSAXParser();
```

```
XMLReader reader = parser.getXMLReader();
```

- ▶ A este *reader* habrá que pasarle el manejador que se encargue de responder a los eventos

```
reader.setContentHandler(new GestionContenido());
```

Acceso a ficheros XML con SAX: lectura

- ▶ Para poder utilizar los métodos que responderán a los eventos, es necesario implementar alguna de las siguientes interfaces:
 - ▶ **ContentHandler**: recibe las notificaciones de los eventos que ocurren en el documento
 - ▶ **DTDHandler**: recoge eventos relacionados con el DTD (*Document Type Definition*)
 - ▶ **ErrorHandler**: define métodos de tratamiento de errores
 - ▶ **EntityResolver**: sus métodos se llaman cuando se encuentra una referencia a una entidad
 - ▶ **DefaultHandler**: proporciona una implementación por defecto para todos los métodos y será necesario desarrollar los que se vayan a utilizar



Acceso a ficheros XML con SAX: lectura

- ▶ Vamos a utilizar **DefaultHandler** e implementar los siguientes métodos:
 - ▶ **startDocument**: se produce al comenzar el procesamiento del documento XML
 - ▶ **endDocument**: se produce al finalizar el procesamiento
 - ▶ **startElement**: se produce al comenzar una etiqueta XML (aquí se leen los atributos de las etiquetas)
 - ▶ **endElement**: al finalizar la etiqueta
 - ▶ **characters**: se produce al encontrar una cadena de texto
- ▶ Por último es necesario pasarle al parser el documento para que se haga el procesamiento:

```
reader.parse(new InputSource(new  
FileInputStream("empleados.xml")));
```

Acceso a ficheros XML con SAX: lectura

- El código de la clase que extiende la interfaz DefaultHandler es:

```
class GestionContenido extends DefaultHandler{

    @Override
    public void startDocument() throws SAXException {
        System.out.println("\nPrincipio del documento XML");
    }

    @Override
    public void endDocument() throws SAXException {
        System.out.println("\nFin del documento XML");
    }

    @Override
    public void startElement(String uri, String localName, String name,
        Attributes attributes) throws SAXException {
        System.out.print("\nEtiqueta -> "+name);
        //Recorremos los atributos
        System.out.println("\t"+attributes.getLength()+" atributos: ");
        for(int i=0;i<attributes.getLength();i++){
            System.out.print("\t"+attributes.getQName(i)+": "+attributes.getValue(i));
        }
    }

    @Override
    public void characters(char[] ch, int start, int length)
        throws SAXException {
        System.out.println("\tTexto: "+String.valueOf(ch, start, length));
    }

    @Override
    public void endElement(String uri, String localName, String name)
        throws SAXException {
        System.out.println("\nfin "+name);
    }
}
```

Acceso a ficheros XML con SAX: escritura

- ▶ Se necesita una clase que derive de **XMLReader** y que implemente una versión del método parser adaptado a nuestros datos.
- ▶ También se necesita un objeto de la clase **InputSource** a la que se le pasará la colección con nuestros datos.
- ▶ Se creará un objeto de tipo Source, al que se le pasarán los dos objetos anteriores, y será el punto de partida de la transformación:

```
XMLReader datosReader = new DatosReader();
```

```
InputSource datosSource = new Alm_datos(50);
```

```
// La clase que contenga los datos deberá extends InputSource
```

```
Source source = new SAXSource(datosReader, datosSource);
```



Acceso a ficheros XML con SAX: escritura

- ▶ Lo siguiente será preparar la salida, con los datos del fichero de salida.

```
File f = new File("empleadosSAX.xml");  
Result resultado = new StreamResult(f);
```

- ▶ Se crea la instancia y se configuran los parámetros del tipo de fichero que se quiere obtener:

```
Transformer transformer =  
TransformerFactory.newInstance().newTransformer();  
transformer.setOutputProperty(OutputKeys.INDENT, "yes");  
transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
```

- ▶ Por último se produce la transformación:

```
transformer.transform(source, resultado);
```

▶

Acceso a ficheros XML con SAX: escritura

```
public class DatosReader implements XMLReader {  
    ..... private ContentHandler handler;  
    ..... private final AttributesImpl atts = new AttributesImpl();  
    ..... //métodos que serán usados por el transformer  
    @Override  
    public void parse(InputSource input) throws IOException, SAXException {  
        if (!(input instanceof Alm_datos)) {  
            String m = "El único parámetro para el parser es un Alm_datos";  
            throw new SAXException(m);  
        }  
        if (handler == null) {  
            throw new SAXException("Sin ContentHandler");  
        }  
        Alm_datos source = (Alm_datos) input;  
        List<Empleado> empleados = source.getDatos();  
        handler.startDocument();  
        handler.startElement("", "empleados", "empleados", atts);  
        for (Empleado empl : empleados) { //El id, se añade como atributo para ver el uso de atributos, podría ser un elemento más  
            atts.addAttribute("", "id", "id", "", String.valueOf(empl.getId()));  
            handler.startElement("", "empleado", "empleado", atts);  
            atts.clear();  
            handler.startElement("", "nombre", "nombre", atts);  
            char[] nombre = empl.getNombre().toCharArray();  
            handler.characters(nombre, 0, nombre.length);  
            .....  
            handler.endElement("", "nombre", "nombre");  
        }  
    }  
}
```

Acceso a ficheros XML con SAX: escritura

//Otros métodos que hay que implementar en DatosReader

.....
@Override

```
public ContentHandler getContentHandler() {  
    return handler; }  
.....
```

@Override

```
public void setContentHandler(ContentHandler handler) {  
    this.handler = handler; }  
.....
```

@Override

```
public void parse(String systemId) { }  
.....
```

@Override

```
public boolean getFeature(String name) {  
    return false; }  
.....
```

@Override

```
public void setFeature(String name, boolean value) { }  
.....
```

@Override

```
public Object getProperty(String name) {  
    return null;  
}  
.....
```

@Override

```
public void setProperty(String name, Object value) { }  
.....
```



Acceso a ficheros XML con SAX: escritura

//Otros métodos que hay que implementar en DatosReader

@Override

```
public void setEntityResolver(EntityResolver resolver) { }
```

@Override

```
public EntityResolver getEntityResolver() {  
    return null; }
```

@Override

```
public void setDTDHandler(DTDHandler handler) { }
```

@Override

```
public DTDHandler getDTDHandler() {  
    return null; }
```

@Override

```
public void setErrorHandler(ErrorHandler handler) { }
```

@Override

```
public ErrorHandler getErrorHandler() {  
    return null;  
}
```

