



Recorriendo XML con SAX

ÍNDICE DE CONTENIDOS

- Características SAX
- Ventajas sobre DOM
- Desventajas sobre DOM
- Ejemplo (Recorrido)

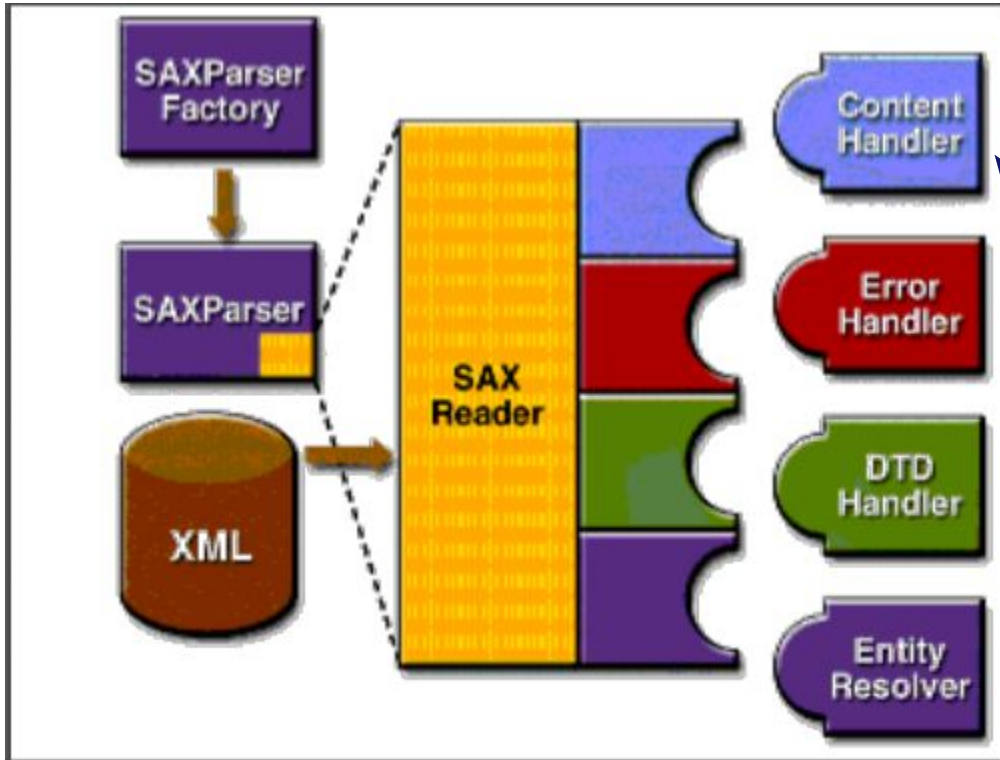


CARACTERÍSTICAS

- Una PASADA por el documento XML.
- Necesito tener el documento completo.
- Basado en eventos y manejadores.
- Los eventos se disparan según el contenido que vamos encontrando



CARACTERÍSTICAS



Interfaces

(que debo implementar en una clase concreta para leer un XML determinado)

NO TENGO QUE IMPLEMENTAR TODOS
DefaultHandler



CARACTERÍSTICAS

Interfaz	Métodos a implementar
ContentHandler	<ul style="list-style-type: none">• startDocument• endDocument• startElement• endElement• characters....
ErrorHandler	<ul style="list-style-type: none">• error• fatalError• warning
DTDHandler	<ul style="list-style-type: none">• notationDecl• unparsedEntityDecl
EntityResolver	<ul style="list-style-type: none">• resolveEntity



VENTAJAS SOBRE DOM

- Menos consumo de memoria (no cargo el documento entero)
- Mejor para documentos grandes



DESVENTAJA SOBRE SAX

- No puedo crear documentos.
- Recorrido solo Top-Down.



EJEMPLO

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<personas>
  <persona id="1">
    <nombre>Pepe</nombre>
    <edad>14</edad>
  </persona>
  <persona id="2">
    <nombre>María</nombre>
    <edad>51</edad>
  </persona>
  <persona id="3">
    <nombre>Raul</nombre>
    <edad>22</edad>
  </persona>
</personas>
```



EJEMPLO

1er Paso - Obtenemos el Parser

```
SAXParserFactory spf = SAXParserFactory.newInstance();  
SAXParser saxParser = spf.newSAXParser();
```



EJEMPLO

2º Paso - Definimos la clase que manejará los eventos

```
public class PersonaSax extends DefaultHandler {
```

*//Esta clase contendrá las implementaciones propias
de los métodos que nos interesen*

```
}
```



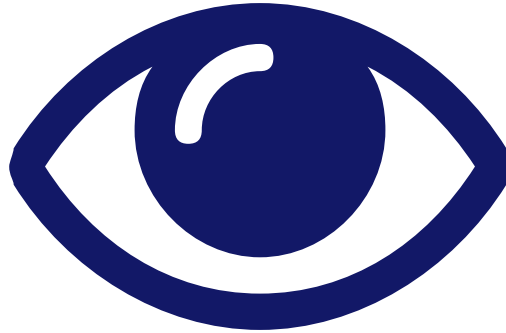
EJEMPLO

3er Paso - Parseamos el Fichero

```
saxParser.parse(new File("personasXML.xml"), personaSax);
```



EJEMPLO



END



prof.jdperez@iesalixar.org

