

Recorriendo XML con SAX

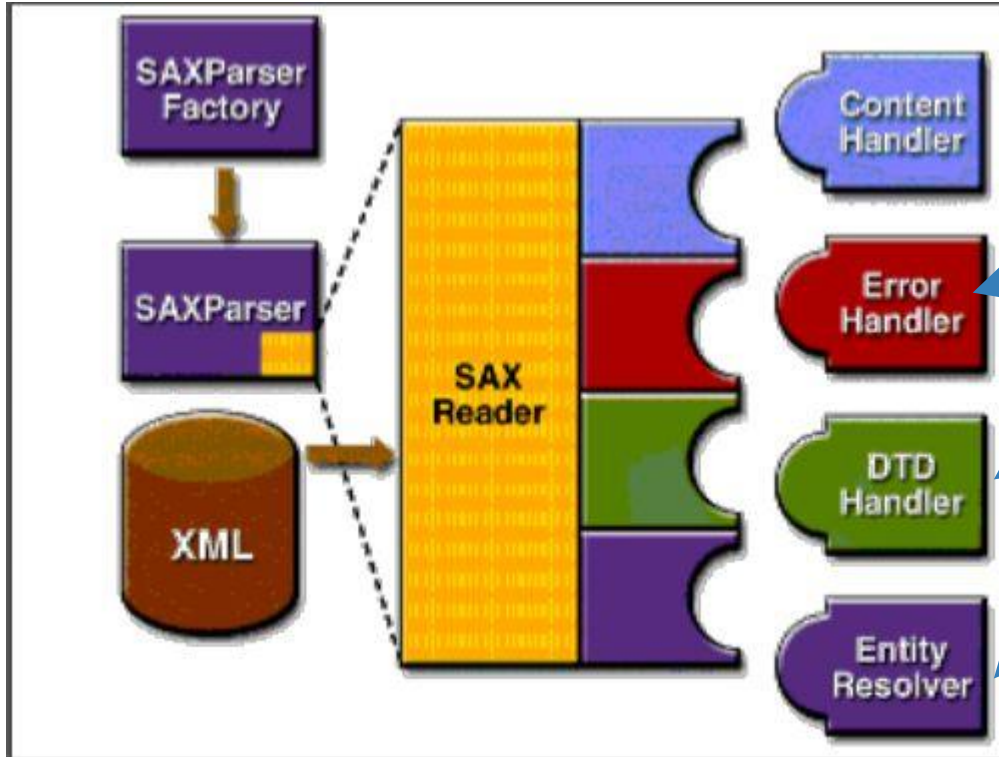
Índice

- **Características SAX**
- **Ventajas/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• unparseEntityDecl
EntityResolver	<ul style="list-style-type: none">• resolveEntity

Ventajas sobre DOM

Ventajas:

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



Desventajas:

- 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);
```

Fin



jgardur081@g.educaand.es