Procesamiento de documentos XML.

Un analizador o parser XML es una herramienta encargada de leer documentos XML, poder acceder a sus elementos y comprobar si el documento es sintácticamente válido. Estas herramientas son módulos, bibliotecas o programas que se ocupan de transformar un archivo XML en una representación interna. Entre esos analizadores o parsers cabe destacar:

- SAX (Simple API for XML): se encarga de recorrer la estructura del documento generando eventos que corresponden a los elementos que se va encontrando.
- DOM (Document Object Model): representa el archivo en una estructura tipo árbol que usará para leer el documento.

SAX y DOM permiten analizar el lenguaje XML y definir la estructura de un documento. La validación del documento XML consiste en comprobar que el documento, además de estar bien formado de acuerdo con las reglas de XML, responde a una estructura definida en una Definición del Tipo de Documento (DTD).

API SAX

Simple API for XML (SAX), es una interfaz simple que se encarga de procesar o analizar la información del documento XML por eventos. SAX lee el documento secuencialmente de principio a fin, sin cargarlo en memoria, de forma que cuando encuentra un elemento se encarga de lanzar su evento asociado. Cuando el evento es lanzado éste puede ser capturado para realizar una función determinada. Esta API está definida en el paquete: javax.xml.parsers.

Para que estos eventos se puedan capturarse y realizar las operaciones que se deseen se debe usar un manejador de eventos. Un manejador es una clase con una serie de métodos y cada método se ejecutará cuando el analizador capture su evento asociado. Estos eventos se producen al leer un documento (al comienzo del documento, apertura o cierre de un elemento, al encontrar una instrucción de proceso o un comentario, etc.).

API DOM

El API "Document Object Model" (DOM) es un conjunto de interfaces que describen una estructura abstracta para un documento XML. DOM carga el documento XML entero en memoria con una estructura tipo árbol. Cada elemento del documento XML se representa con un nodo (DOMNode). DOM está definido en los paquetes org.w3c.dom y javax.xml.parsers.

El árbol jerárquico de información en memoria permite que a través del manejador pueda manipularse la información: crear o eliminar información de un nodo en cualquier punto del árbol, acceder o cambiar su contenido y mover la herencia de nodos.

Para poder hacer uso del parser se debe obtener una instancia de una factoría analizadora (DocumentBuiderFactory). Con esta factoría se crea el analizador (DocumentBuilder) que es capaz de producir un nodo Document que cumple la especificación DOM, es decir, crear el inicio del árbol. Se puede crear un nodo Document vacío con el método newDocument() o crear el árbol completo de un documento XML pasándole éste al analizador con el método parser (documento_XML). A este analizador se le asocia/n el/los manejador/es, que indicarán las operaciones a realizar al capturar un evento lanzado por el analizador. Estas operaciones a realizar se encuentran definidas en sus métodos. Y, por último, se le pasa al analizador el documento para empezar a leer el documento y validarlo

Las características principales de la API DOM son las siguientes:

- DOM representa en memoria el documento XML mediante una estructura tipo árbol.
- Cada elemento del documento XML se representa con un nodo dentro del árbol.
- Se proporcionan gran variedad de funciones para navegar a través del árbol DOM.
- Permite manipular el árbol en memoria, añadiendo un nuevo elemento o eliminando uno existente, actualizándolo o únicamente consultarlo.
- Permite la validación de un documento XML.

DOM permite disponer de la estructura del documento XML en memoria, luego es apropiado para el manejo de documentos XML que no sean de gran tamaño, ya que implicaría un gasto de memoria considerable. Está orientado a aplicaciones en las que se quiere consultar el documento varias veces o incluso modificarlo gracias, también, a que el árbol permanece en memoria. Sin embargo, hay que tener en cuenta que el almacenamiento del documento XML en memoria mediante la estructura en árbol requiere de un coste en tiempo adicional además del coste en memoria.

Diferencias entre SAX y DOM:

SAX	DOM
Modelo basado en eventos	Estructura de datos tipo árbol
Acceso serie (flujo de eventos)	Acceso aleatorio (estructura de datos en memoria)
Bajo uso de memoria (sólo se generan eventos)	Alto uso de memoria (todo el documento se carga en memoria)
Para procesar partes del documento (capturar eventos importantes)	Para editar el documento (procesar la estructura de datos en memoria)
Para procesar el documento sólo una vez (flujo de eventos temporal).	Para procesar el documento múltiples veces (documento cargado en memoria).

Utilidad de cada uno de los analizadores:

SAX	DOM
Cuando no haya una modificación estructural del documento.	Para modificar el documento.
Menor gasto de memoria y mayor rapidez.	Si se necesita realizar múltiples procesados.
Si sólo se necesitan partes de documentos	Evita tener que volver a analizar el documento
Para documentos XML grandes, en donde sólo haya que procesar una pequeña parte de la información.	Para documentos XML pequeños que necesiten ser procesados en su práctica totalidad.
Permite recorrer secuencialmente un documento XML y responder a una serie de eventos.	Evita tener que construir tu propio árbol.