

# APACHE XERCES STAX PARSER - OVERVIEW

[http://www.tutorialspoint.com/xerces/xerces\\_stax\\_parser.htm](http://www.tutorialspoint.com/xerces/xerces_stax_parser.htm)

Copyright © tutorialspoint.com

StAX is a JAVA based API to parse XML document in a similar way as SAX parser does. But there are two major difference between the two APIs

- StAX is a PULL API where as SAX is a PUSH API. It means in case of StAX parser, client application need to ask StAX parser to get information from XML whenever it needs but in case of SAX parser, client application is required to get information when SAX parser notifies the client application that information is available.
- StAX API can read as well as write XML documents. Using SAX API, xml can be only be read.

Following are the features of StAX API

- Reads an XML document from top to bottom, recognizing the tokens that make up a well-formed XML document
- Tokens are processed in the same order that they appear in the document
- Reports the application program the nature of tokens that the parser has encountered as they occur
- The application program provides an "event" reader which acts as an iterator and iterates over the event to get the required information. Another reader available is "cursor" reader which acts as a pointer to xml nodes.
- As the events are identified, xml elements can be retrieved from the event object and can be processed further.

## When to use?

You should use a StAX parser when:

- You can process the XML document in a linear fashion from the top down.
- The document is not deeply nested.
- You are processing a very large XML document whose DOM tree would consume too much memory. Typical DOM implementations use ten bytes of memory to represent one byte of XML.
- The problem to be solved involves only part of the XML document.
- Data is available as soon as it is seen by the parser, so StAX works well for an XML document that arrives over a stream.

## Disadvantages of SAX

- We have no random access to an XML document since it is processed in a forward-only manner
- If you need to keep track of data the parser has seen or change the order of items, you must write the code and store the data on your own

## XMLEventReader Class

This class provide iterator of events which can be used to iterate over events as they occur while parsing the XML document

- **StartElement asStartElement** - used to retrieve value and attributes of element.
- **EndElement asEndElement** - called at the end of a element.
- **Characters asCharacters** - can be used to obtain characters such a CDATA, whitespace etc.

## XMLEventWriter Class

This interface specifies methods for creating an event.

- **addEvent** - Add event containing elements to XML.

## XMLStreamReader Class

This class provide iterator of events which can be used to iterate over events as they occur while parsing the XML document

- **int next** - used to retrieve next event.
- **boolean hasNext** - used to check further events exists or not
- **String getText** - used to get text of an element
- **String getLocalName** - used to get name of an element

## XMLStreamWriter Class

This interface specifies methods for creating an event.

- **writeStartElementStringlocalName** - Add start element of given name.
- **writeEndElementStringlocalName** - Add end element of given name.
- **writeAttributeStringlocalName Stringvalue** - Write attribute to an element.

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js