Java XPath Parser - Overview

XPath is an XML Path language to find information in an XML file. It is an official recommendation of the World Wide Web Consortium (W3C). It is used to traverse elements and attributes of an XML document. XPath provides various types of expressions which can be used to enquire relevant information from the XML document and it is mainly used in XSLT standard.

XPath Terminology

- **Structure Definitions** – XPath defines the parts of an XML document like element, attribute, text, namespace, processing-instruction, comment, and document nodes.
- **Path Expressions** – XPath provides powerful path expressions such as select nodes or list of nodes in XML documents.
- **Standard Functions** – XPath provides a rich library of standard functions for manipulation of string values, numeric values, date and time comparison, node and QName manipulation, sequence manipulation, Boolean values, etc.
- **Axes** – XPath has thirteen different axes that retrieves the relative elements of the current element such as ancestor, child, descendant, preceding, following etc.

XPath Expressions

XPath uses a path expression to select node or list of nodes from an XML document. Following is a list of useful path expressions to select any node/list of nodes from an XML document.

Expression	Description
node-name	Selects all nodes with the given "nodename"
/	Selection starts from the root node
//	Selection starts from the current node that match the selection
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

student	Selects all nodes with the name "student"
class/student	Selects all student elements that are children of class
//student	Selects all student elements no matter where they are in the document

Expressions with Predicates

XPath expressions can be used along with predicates to obtain a specific node or a node containing specific value and are defined using [...].

Expression	Result
/class/student**[1]**	Selects the first student element that is the child of the class element.
/class/student**[last()]**	Selects the last student element that is the child of the class element.
/class/student**[last()-1]**	Selects the last but one student element that is the child of the class element.
//student**[@rollno = '493']**	Selects all the student elements that have an attribute named rollno with a value of '493'