

JAVA XML - INTERVIEW QUESTIONS

http://www.tutorialspoint.com/java_xml/java_xml_interview_questions.htm

Copyright © tutorialspoint.com

Dear readers, these **JAVA based XML Parsing Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **JAVA based XML Parsing**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer –

What XML stands for?

XML stands for Extensible Markup Language.

What are the advantages of using XML?

Following are the advantages that XML provides –

- **Technology agnostic** - Being plain text, XML is technology independent. It can be used by any technology for data storage and transmission purpose.
- **Human readable**- XML uses simple text format. It is human readable and understandable.
- **Extensible** - in XML, custom tags can be created and used very easily.
- **Allow Validation** - Using XSD, DTD and XML structure can be validated easily.

What are the disadvantages of using XML?

Following are the disadvantages of XML usage –

- **Redundant Syntax** - Normally XML file contains lot of repetitive terms.
- **Verbose**-Being a verbose language, XML file size increases the transmission and storage costs.

What is XML Parsing?

Parsing XML refers to going through XML document to access data or to modify data in one or other way.

What is XML Parser?

XML Parser provides way how to access or modify data present in an XML document. Java provides multiple options to parse XML document.

Name some of the parsers which are commonly used to parse XML documents.

Following are various types of parsers which are commonly used to parse XML documents –

- **Dom Parser** - Parses the document by loading the complete contents of the document and creating its complete hierarchical tree in memory.
- **SAX Parser** - Parses the document on event based triggers. Does not load the complete document into the memory.
- **JDOM Parser** - Parses the document in similar fashion to DOM parser but in more easier way.
- **StAX Parser** - Parses the document in similar fashion to SAX parser but in more efficient way.
- **XPath Parser** - Parses the XML based on expression and is used extensively in conjunction with XSLT.
- **DOM4J Parser** - A java library to parse XML, XPath and XSLT using Java Collections Framework , provides support for DOM, SAX and JAXP.

What DOM stands for?

DOM stands for Document Object Model.

What is DOM?

DOM stands for Document Object Model and it is an official recommendation of the World Wide Web Consortium W3C. It defines an interface that enables programs to access and update the style, structure, and contents of XML documents. XML parsers that support the DOM implement that interface.

When to use a DOM Parser?

You should use a DOM parser when –

- You need to know a lot about the structure of a document
- You need to move parts of the document around *you might want to sort certain elements, for example*
- You need to use the information in the document more than once

What DOM Parser returns?

When you parse an XML document with a DOM parser, you get back a tree structure that contains all of the elements of your document. The DOM provides a variety of functions you can use to examine the contents and structure of the document.

What are the advantages of DOM Parsing?

The DOM is a common interface for manipulating document structures. One of its design goals is that Java code written for one DOM-compliant parser should run on any other DOM-compliant parser without changes.

What are the key components/interfaces of DOM Parsing?

The DOM defines several Java interfaces. Here are the most common interfaces –

- **Node** - The base datatype of the DOM.
- **Element** - The vast majority of the objects you'll deal with are Elements.
- **Attr** Represents an attribute of an element.
- **Text** The actual content of an Element or Attr.
- **Document** Represents the entire XML document. A Document object is often referred to as a DOM tree.

Name some of the important DOM parsing methods.

When you are working with the DOM, there are several methods you'll use often –

- **Document.getDocumentElement** - Returns the root element of the document.
- **Node.getFirstChild** - Returns the first child of a given Node.
- **Node.lastChild** - Returns the last child of a given Node.
- **Node.getNextSibling** - These methods return the next sibling of a given Node.
- **Node.getPreviousSibling** - These methods return the previous sibling of a given Node.
- **Node.getAttribute** *attrName* - For a given Node, returns the attribute with the requested name.

Can we create an XML document using DOM parser?

Yes! Using DOM parser, we can parse, modify or create a XML document.

What SAX stands for?

SAX stands for Simple API for XML.

What is a SAX Parser?

SAX Parser is an event-based parser for xml documents.

How a SAX Parser works?

SAX *theSimpleAPIforXML* is an event-based parser for xml documents. Unlike a DOM parser, a SAX parser creates no parse tree. SAX is a streaming interface for XML, which means that applications using SAX receive event notifications about the XML document being processed an element, and attribute, at a time in sequential order starting at the top of the document, and ending with the closing of the ROOT element.

When to use a SAX Parser?

You should use a SAX 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 SAX works well for an XML document that arrives over a stream

What are the disadvantages of SAX Parsing?

- 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

Name some of the important SAX parsing methods.

ContentHandler Interface specifies the callback methods that the SAX parser uses to notify an application program of the components of the XML document that it has seen.

- **void startDocument** - Called at the beginning of a document.
- **void endDocument** - Called at the end of a document.
- **void startElementStringuri, StringlocalName, StringqName, Attributesatts** - Called at the beginning of an element.
- **void endElementStringuri, StringlocalName, StringqName** - Called at the end of an element.
- **void characterschar[]ch, intstart, intlength** - Called when character data is encountered.
- **void ignorableWhitespacechar[]ch, intstart, intlength** - Called when a DTD is present and ignorable whitespace is encountered.
- **void processingInstructionStringtarget, Stringdata** - Called when a processing instruction is recognized.
- **void setDocumentLocatorLocatorlocator)** - Provides a Locator that can be used to identify positions in the document.
- **void skippedEntityStringname** - Called when an unresolved entity is encountered.

- **void startPrefixMapping***Stringprefix, Stringuri* - Called when a new namespace mapping is defined.
- **void endPrefixMapping***Stringprefix* - Called when a namespace definition ends its scope.

Name some methods for processing the attributes connected to an element in SAX parsing.

Attributes Interface specifies methods for processing the attributes connected to an element.

- **int getLength** - Returns number of attributes.
- **String getQName***intindex*
- **String getValue***intindex*
- **String getValue***Stringqname*

Can we create an XML document using SAX parser?

No! Using SAX parser, we can only parse or modify a XML document.

What is JDOM Parser?

JDOM is an open source, java based library to parse XML document and it is typically java developer friendly API.

What are the benefits of JDOM parser?

It is java optimized, it uses java collection like List and Arrays. It works with DOM and SAX APIs and combines the best of the two. It is of low memory footprint and is nearly as fast as SAX.

When to use a JDOM Parser?

You should use a JDOM parser when –

- You need to know a lot about the structure of a document.
- You need to move parts of the document around *you might want to sort certain elements, for example.*
- You need to use the information in the document more than once.
- You are a java developer and want to leverage java optimized parsing of XML.

What are the advantages of JDOM parser?

When you parse an XML document with a JDOM parser, you get the flexibility to get back a tree structure that contains all of the elements of your document without impacting the memory footprint of the application. The JDOM provides a variety of utility functions you can use to examine the contents and structure of the document in case document is well structured and its structure is known.

JDOM gives java developers flexibility and easy maintainability of xml parsing code. It is light weight and quick API.

Name some of the important JDOM classes.

The JDOM defines several Java classes. Here are the most common classes –

- **Document** - Represents the entire XML document. A Document object is often referred to as a DOM tree.
- **Element** - Represents an XML element. Element object has methods to manipulate its child elements, its text, attributes and namespaces.
- **Attribute** Represents an attribute of an element. Attribute has method to get and set the value of attribute. It has parent and attribute type.
- **Text** Represents the text of XML tag.

- **Comment** Represents the comments in a XML document.

Name some of the important JDOM methods.

When you are working with the JDOM, there are several methods you'll use often –

- **SAXBuilder.buildxmlSource** - Build the JDOM document from the xml source.
- **Document.getRootElement** - Get the root element of the XML.
- **Element.getName** - Get the name of the XML node.
- **Element.getChildren** - Get all the direct child nodes of an element.
- **Node.getChildrenName** - Get all the direct child nodes with a given name.
- **Node.getChildName** - Get first child node with given name.

Can we create an XML document using JDOM parser?

Yes! Using JDOM parser, we can parse, modify and create a XML document.

What is a StAX Parser?

StAX is a JAVA based API to parse XML document in a similar way as SAX parser does but 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.

Can we create an XML document using StAX parser?

Yes! Using StAX parser, we can parse, modify and create a XML document.

Is StAX parser a PULL API?

Yes! StAX is a PULL API.

When you should use a StAX parser?

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.

What are the disadvantages of StAX parser?

- 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

Explain XMLEventReader Class of StAX parser.

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.

Explain XMLEventWriter Class of StAX parser.

This interface specifies methods for creating an event.

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

Explain XMLStreamReader Class of StAX parser.

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

Explain XMLStreamWriter Class of StAX parser.

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.

What is XPath?

The XPath is an official recommendation of the World Wide Web Consortium W3C. It defines a language to find information in an XML file. It is used to traverse elements and attributes of an XML document. XPath provides various type of expressions which can be used to enquire relevant information from the XML document.

What are the key components of XPath?

Following are the key components of XPath –

- **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 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.
- **Major part of XSLT** XPath is one of the major element in XSLT standard and is must have knowledge in order to work with XSLT documents.
- **W3C recommendation** XPath is official recommendation of World Wide Web Consortium W3C.

What is predicate in XPath?

Predicate are used to find 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'

What is path expression in XPath?

XPath uses a path expression to select node or list of nodes from an xml document. Following is the list of useful paths and expression to select any node/ list of nodes from an xml document.

Expression	Description
node-name	Select all nodes with the given name "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	Example – Selects all nodes with the name "student"
class/student	Example: Selects all student elements that are children of class
//student	Selects all student elements no matter where they are in the document

Can we create an XML document using XPath parser?

No! XPath parser is used to to navigate XML Document only. It is better to use DOM parser for creating XML.

What is DOM4J Parser?

DOM4J is an open source, java based library to parse XML document and it is highly flexible, high-performance, and memory-efficient API. It is java optimized, it uses java collection like List and Arrays. It works with DOM, SAX, XPath and XSLT. It can parse large XML document with very low memory footprint.

When to use a DOM4J Parser?

You should use a DOM4J parser when –

- You need to know a lot about the structure of a document
- You need to move parts of the document around *you might want to sort certain elements, for example*
- You need to use the information in the document more than once
- You are a java developer and want to leverage java optimized parsing of XML.

When are the benefits of using a DOM4J Parser?

When you parse an XML document with a DOM4J parser, you get the flexibility to get back a tree structure that contains all of the elements of your document without impacting the memory footprint of the application. The DOM4J provides a variety of utility functions you can use to examine the contents and structure of the document in case document is well structured and its structure is known. DOM4J uses XPath expression to navigate through the XML document.

When are the advantages of using a DOM4J Parser?

DOM4J gives java developers flexibility and easy maintainability of xml parsing code. It is light weight and quick API.

Name some of the important DOM4J classes.

The DOM4J defines several Java classes. Here are the most common classes –

- **Document** - Represents the entire XML document. A Document object is often referred to as a DOM tree.
- **Element** - Represents an XML element. Element object has methods to manipulate its child elements, its text, attributes and namespaces.
- **Attribute** Represents an attribute of an element. Attribute has method to get and set the value of attribute. It has parent and attribute type.
- **Node** Represents Element, Attribute or ProcessingInstruction

Name some of the important DOM4J methods.

When you are working with the DOM4J, there are several methods you'll use often –

- **SAXReader.readxmlSource** - Build the DOM4J document from the xml source.
- **Document.getRootElement** - Get the root element of the XML.
- **Element.nodeindex** - Get the XML node at particular index in the element.
- **Element.attributes** - Get all the attributes of an element.
- **Node.valueOf@Name** - Get the value of an attribute with given name of the element.

Can we create an XML document using DOM4J parser?

Yes! Using DOM4J parser, we can parse, modify and create a XML document.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

Processing math: 98%