Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › member-paid › XML Processing Interview Q&A

XML Processing Interview Q&A

Posted on September 14, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

0 Like Share Tweet 0 G+1 Share

Q1. What libraries can you use in Java to process XML documents?

A1.

#1. DOM Parser loads the whole XML structure into memory, and you can read and write. This is useful for smaller XML documents which don't impact memory usage. JDOM is another alternative to DOM4J.

#2. SAX Parser to only read an XML document. Good to parse large XML documents. SAX 'pushes' XML events, leaving it up to you to determine where the XML events belong in your code logic.

#3. StAx Reader/Writer works with a datastream oriented interface. The program asks for the next element when it's

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all

- Ice Breaker Interview
- Core Java Interview (
- JEE Interview Q&A (3
- Pressed for time? Jav
- SQL, XML, UML, JSC
  - ERD (1)
  - NoSQL (2)
  - Regex (2)
  - SQL (7)
  - UML (1)
  - JSON (2)
  - XML (2)
    - XML basics inter
    - XML Processing
  - XSD (2)
  - YAML (2)
- Hadoop & BigData Int
- Java Architecture Inte
- Scala Interview Q&As
- Spring, Hibernate, & I
- Testing & Profiling/Sa

ready just like a cursor/iterator. StAX 'pulls' XML events, leaving it up to you to determine where in your program / data to receive the XML events. In general, StAX is as effecient as SAX.

#4. JAXB for object to XML and XML to Object mapping. Very widely used in Web services. The POJOs are annotated to map XML or JSON to object and vice versa. **MOXy** implementation of JAXB is powerful as it supports XPath for mapping, and can be used for both XML and JSON.

Java 6 onwards, there is built in support for all the different types of parses listed above. In general, will be favoring **StAX**, and move to SAX if performance is a real concern. When using OXM (Object To Xml Mapping) then **JAXB** is used.

Q2. What is a StAX Parser and when will you use it?
A2. The StAX Java API for XML processing is designed for parsing XML streams, just like the SAX API's, but

– StAX is a "**pull**" API whereas SAX is a "push" API.
– StAX can do both **XML reading** and **writing** whereas SAX can only do XML reading.

Q3. Why use StAX when there is DOM?
A3. The **DOM parsing involves creating in-memory objects** representing an entire document tree for an XML document. Once in memory, DOM trees can be navigated and parsed arbitrarily, providing the maximum flexibility for developers. However this flexibility comes at a cost of large memory footprint and significant processor requirements, as the entire representation of the document must be held in memory as objects for the duration of the document processing. This may not be an issue when working with small documents, but memory and processor requirements can escalate quickly for larger size documents.

StAX (Streaming Api for XML) involves streaming where streaming refers to a programming model where XML data sets are transmitted and parsed serially at application runtime

⊞ Other Interview Q&A f
⊞ ▶Free Java Interviev

16 Technical Key Areas

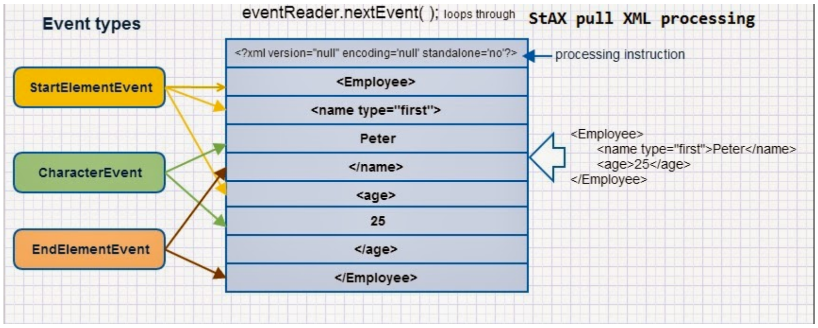
open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience I
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2
⊞ Hadoop & Spark Tuto
⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tu
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

as events like StartElementEvent, CharacterEvent, EndElementEvent, etc, often in real time and dynamically where contents are not precisely known beforehand. These stream-based parsers can start generating output immediately, and data set elements can be discarded and garbage collected immediately after they are used. This ensures smaller memory footprint, reduced processor requirements, and higher performance in certain scenarios. This memory and processing efficiency comes at the cost where streaming can only see the data set state at one location at a time in the document.

Here is how the events are sequentially processed:



```
<Employee>
    <name type="first">Peter</name>
    <age>25</age>
</Employee>
```

StAX Parser

Q4. What is the major advantage of StAX over SAX?
A4. The major advantage of StAX over SAX is that the pull model allows sub parsing of the XML input. You can extract out the element name, then the attributes, and then the characters (i.e. content).

Q5. What does JAXB stand for? What is an XML Binding?
A5. JAXB means Java API for Xml Binding. XML binding maps an XML to in-memory Java objects. The principle advantage of using JAXB when marshalling and demarshalling XML is that is simplifies the programming model by allowing us to simply annotate a few POJOs and use the JAXB API's and you can serialise to XML and deserialise from XML very easily. This makes it much simpler than alternatives such as DOM and SAX.

100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code? (
⊞ ♦ Complete the given
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

How good are your .....?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum

Q6. Why use JAXB?

A6. JAXB is a reference interface for which you can have a number of different implementations like default reference implementation shipped JDK6, JaxMeAPI, **MOXy**, Metro, etc. JAXB is available in JDK6 onwards, so it doesn't require any external library and it doesn't require a XML schema to work. XML schema is optional. Easy to use as you can use annotations on your model classes. It is supported by RESful frameworks like Jersey, Spring MVC, RESTEasy, etc.

Q7. Why would be the motivating factor to use MOXy implementation of JAXB as opposed to the default implementation provided by the JDK6 implementation?

A7. MOXy supports **@XmlPath** annotation and other extensions will make your implementation cleaner.

1. @XmlPath extension which is inspired from XPath
2. The Jersey JAX-RS reference implementation provides JSON binding via MOXy.

Popular Posts

♦ 11 Spring boot interview questions & answers

828 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

768 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

389 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

296 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

286 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

280 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

240 views

001B: ♦ Java architecture & design concepts interview questions & answers

202 views

Bio | **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ JavaScript mistakes interview Q&A JSON interview Q&A ›

Posted in member-paid, XML

Tags: Architect FAQs, Java/JEE FAQs

Leave a Reply

Logged in as geethika. Log out?

Comment

Post Comment

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from "Blah blah" to "Wow"? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑