XML basics interview Q&A

Posted on September 13, 2014 by Arulkumaran Kumaraswamipillai — No Comments ↓

Q1. What is an XML?

A1. XML stands for e**X**tensible **M**arkup **L**anguage. XML is a grammatical system for constructing custom markup languages for describing business data, mathematical data, chemical data etc. XML loosely couples disparate applications or systems utilizing JMS, Web services etc. XML uses the same building blocks that HTML does: elements, attributes and values.

Q2. Why is XML popular and important?

A2. XML is protocol and language neutral.

Scalable: Since XML is not in a binary format you can create and edit files with anything and it's also easy to debug. XML can be used to efficiently store small amounts of data like configuration files (web.xml, application.xml, struts-config.xml,

etc) to large company wide data with the help of XML stored in the database.

Fast Access: XML documents benefit from their hierarchical structure. Hierarchical structures are generally faster to access because you can drill down to the section you are interested in.

Easy to identify and use: XML not only displays the data but also tells you what kind of data you have. The **mark up tags** identifies and groups the information so that different information can be identified by different application.

Stylability: XML is style-free and whenever different styles of output are required the same XML can be used with different style-sheets (XSL) to produce output in XHTML, PDF, TEXT, another XML format, etc.

Linkability, in-line usability, universally accepted standard with free/inexpensive tools etc.

Q3. When would you not use an XML?
A3. XML is verbose and it can be 4-6 times larger in size compared JSON or csv data. If your network lacked bandwidth and/or your content is too large and network throughput is vital to the application then you may consider using JSON or csv data instead of an XML. If your end points support JSON, it might be a better alternative if markup is not required.

Q4. How would you decide to store data as elements or as attributes?
A4. A question arising in the mind of XML designers is whether to model and encode certain information using an element, or alternatively, using an attribute. The answer to the above question is not clear-cut. But the general guideline is:

Using an element:: If you consider the information in question to be part of the essential material that is being expressed or communicated in the XML, put it in an element.

⊞ Other Interview Q&A t
⊞ ▶ 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 In
⊞ 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)

```
1 <book><title>Lord of the Rings</title></book>
```

Using an attribute:: If you consider the information to be peripheral or incidental to the main communication, or purely intended to help applications process the main communication, use attributes.

```
1 <book title="Lord of the Rings"/>
```

The principle is **data goes in elements** and **metadata goes in attributes**. Elements are also useful when they contain special characters like “<”, “>”, etc which are harder to use in attributes. The most important reason to use element is its **extensibility**. It is far easier to create child elements to reflect complex content than to break an attribute into pieces. You can use attributes along with elements to refine your understanding of that element with extra information. Attributes are **less verbose** but using attributes instead of child elements with the view of optimizing document size is a short term strategy, which can have long term consequences.

Q5. What is XSD? How does it differ from DTD?
A5. XSD stands for **X**ml **S**chema **D**efinition, which is a successor of DTD (Document Type Definition). So XSD is a building block of an XML document. If you have DTD then why use XSD you may ask?

XSD is more powerful and extensible than DTD. XSD has:

— Support for simple and complex data types.
— Uses XML syntax. So XSD are extensible just like XML because they are written in XML.
— Better data communication with the help of data types. For example a date like 03-04-2005 will be interpreted in some countries as 3rd of April 2005 and in some other countries as 04th March 2005.

```
1 <?xml version="1.0"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLS
3 <xs:element name="note">
```

```
    <xs:complexType>
        <xs:sequence>
           <xs:element name="to" type="xs:string
           <xs:element name="from" type="xs:stri
           <xs:element name="title" type="xs:str
           <xs:element name="content" type="xs:s
        </xs:sequence>
     </xs:complexType>
   <xs:attribute name="language" type=”xs:string
</xs:element>

</xs:schema>
```

Q6. What is an XSL?

A6. XSL stands for e**X**tensible **S**tylesheet **L**anguage. The XSL consists of 3 parts:

1) **XSLT:** Language for transforming XML documents from one to another.

2) **XPath:** Language for defining the parts of an XML document.

3) **XSL-FO:** Language for formatting XML documents. For example to convert an XML document to a PDF document etc.

XSL can be thought of as a set of languages that can :

— Define parts of an XML.
— Transform an XML document to XHTML (eXtensible Hyper Text Markup Language) document.
— Convert an XML document to a PDF document.
— Filter and sort XML data.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/
  <xsl:template match="/">
    <xsl:apply-templates select="note " />
  </xsl:template>

  <xsl:template match="note">
    <html>
      <head>
        <title><xsl:value-of select="content/@la
        </title>
      </head>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Q7. What is an XPath?

A7. **X**ml **Path** Language, a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer. We can write both the patterns (context-free) and expressions using the XPath Syntax. XPathis also used in XQuery.

```
1 <xsl:template match="content[@language='English']
2 .....
3 <td><xsl:value-of  select="content/@language" /><
```

Q8. What is version information in XML?

A8. Version information in an XML is a processing instruction.

```
1 <?xml version="1.0" ?>
```

Tags that begin and end with "?" are called processing instructions. The processing instructions can also be used to call a style sheet for an XML as shown below:

```
1 <?xml-stylesheet type="text/css" href="MyStyle.cs
```

Q9. What is a CDATA section in an XML?

A9. If you want to write about elements and attributes in your XML document then you will have to prevent your parser from interpreting them and just display them as a regular text. To do this, you must enclose such information in a CDATA section.

```
1 <![CDATA[ <customername id="123" > John </custome
```

Q10. How will you embed an XML content within an XML document?

A10. By using a CDATA section.

```
1  <message>
2          <from>LoansSystem</from>
3          <to>DocumentSystem</to>
4          <body>
```

```
                <![CDATA[
                        <application>
                                <number>456</number
                                <name>Peter</name>
                                <detail>blah blah</
                        </application>
                ]]>
        </body>
</message>

```

Q11. How do you write comments in an XML document?
A11.

```
<!--  This is an XML  comment -->
```

Q12. How do you write an attribute value with single quotes? How do you write an element value of “> 500.00”?
A12. You need to use an internal entity reference like < for <, > for >, & for &, " for “, ' for ‘.

```
<customer name=”&quot;Mr. Smith&quot;” />
<cost> &gt; 500.00</cost>

```

Q13. What is a well-formed XML document?
A13. A well formed document adheres to the following rules for writing an XML.

— A root element is required. A root element is an element, which completely contains all the other elements.
— Closing tags are required. abc or
— Elements must be properly nested.
— XML is case sensitive. and elements are considered completely separate.
— An attribute’s value must always be enclosed in either single or double quotes.
— Entity references must be declared in a DTD before being used except for the 5 built-in (<, > etc) discussed in the previous question.

Q14. What is a valid XML document?
A14. For an XML document to be valid, it must conform to the rules of the corresponding DTD (Document Type

Definition – internal or external) or XSD (XML Schema Definition).

Q15. How will you write an empty element?
A15.

```
1 <name age="25"></name>
```

or

```
1 <name age="25" />
```

Q16. What is a namespace in an XML document?
A16. Namespaces are used in XML documents to distinguish one similarly titled element from another. A namespace must have an absolutely unique and permanent name. In an XML, name space names are in the form of a URL. A default namespace for an element and all its children can be declared as follows:

```
1 <accounts xmlns="http://www.bank1.com/ns/account"
2     …
3 </accounts>
```

Individual elements can be labeled as follows:

```
1 <accounts xmlns="http://www.bank1.com/ns/account"
2       <name>FlexiDirect</name>  <!-- uses the defa
3       <bank2:name>Loan</bank2:name> <!-- uses the
4       …
5 </accounts>
6
```

Q17. Explain where your project needed XML documents?
A17. It is hard to find a project, which does not use XML documents.

— XML is used to communicate with disparate systems via messaging or Web Services.
— XML based protocols and standards like SOAP, ebXML,

WSDL etc are used in Web Services.
— XML based deployment descriptors like web.xml, ejb-jar.xml, etc are used to configure the JEE containers.
— XML based configuration files are used by open-source frameworks like Hibernate, Spring, and Struts to name a few.

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**

‹ Security holes & how to fix them interview Q&A

JavaScript Vs Java interview Q&A ›

Posted in member-paid, XML

Tags: 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

↑