Top 10 Interview Questions for DITA Developers

1. TOP 10 INTERVIEW QUESTIONS FOR DITA DEVELOPERS

As DITA becomes more pervasive, hiring managers won't have to look very hard to find candidates claiming to have experience working on projects involving DITA. Despite this trend, it is still not an easy task to find a truly skilled XML developer. This fact, combined with the increasing compensation awarded to job candidates, makes hiring the right people one of the most important parts of any IT project. Consequently, the list of questions below is intended to be a guide for managers faced with the task of filling positions within their organizations that require a solid understanding of the foundations of XML-related technologies.

1.Describe the differences between XML and HTML.

It's amazing how many developers claim to be proficient programming with XML, yet do not understand the basic differences between XML and HTML. Anyone with a fundamental grasp of XML should be able describe some of the main differences outlined in the table below:

Differences Between XML and HTML

XML	HTML
User definable tags	Defined set of tags designed for web display
Content driven	Format driven
End tags required for well formed documents	End tags not required
Quotes required around attributes values	Quotes not required
Slash required in empty tags	Slash not required

2. Describe the role that XSL can play when dynamically generating HTML pages from a relational database.

Even if candidates have never participated in a project involving this type of architecture, they should recognize it as one of the common uses of XML. Querying a database and then formatting the result set so it can be validated as an XML document allows developers to translate the data into an HTML table using XSLT rules. Consequently, the format of the resulting HTML table can be modified without changing the database query or application code since the document rendering logic is isolated to the XSLT rules.

3. Give a few examples of types of applications that can benefit from using XML.

There are literally thousands of applications that can benefit from XML technologies. The point of this question is not to have the candidate rattle off a laundry list of projects that they have worked on, but rather explain the rationale for choosing XML using a few real world examples.For instance, one appropriate answer is XML allows content management systems to store documents independently of their format thereby reducing data redundancy. Another answer relates to B2B exchanges or supply chain management systems. In these instances, XML provides a mechanism for multiple companies to exchange data according to an agreed upon set of rules. A third common response involves wireless applications that require WML to render data on hand held devices.

4. What is DOM and how does it relate to XML?

The Document Object Model (DOM) is an interface specification maintained by the W3C DOM Workgroup that defines application independent mechanism to access, parse or update XML data. It simple terms it is hierarchical model that allows developers to easily manipulate XML documents. Any developer that has worked extensively with XML should be able to discuss the concept and use of DOM objects freely. Additionally, it is not unreasonable to expect advanced candidates to thoroughly understand its internal workings and be able to explain how DOM differs from event based interface specifications such as SAX.

5. What is SOAP and how does it relate to XML?

The Simple Object Access Protocol (SOAP) uses XML to define a protocol for the exchange of information in distributed computing environments. SOAP consists of three components: an envelope, a set of encoding rules, and a convention for representing remote procedure calls. Unless experience with SOAP is a direct requirement for the open position, knowing

Top 10 Interview Questions for DITA Developers

the specifics of the protocol or how it can be used in conjunction with HTTP is not as important as identifying it as a natural application of XML.

6. Walk through the steps necessary to parse XML documents?

Superficially, this is a fairly basic question. However, the point is not to determine whether candidates understand the concept of a parser but rather have them walk through the process of parsing XML documents step-by-step. Determining whether a non-validating or validating parser is needed, choosing the appropriate parser and handling errors are all important aspects to this process that should be included in the candidate's response.

7. Give some examples of XML DTD's or schemas that you have worked with.

Although XML does not require data to be validated against a DTD, many of the benefits of using the technology are derived from being able to validate XML documents against business or technical architecture rules. Polling for the list of DTD's that developers have worked with provides insight to their general exposure to the technology. The ideal candidate will have knowledge of several of the commonly used DTD's such as FpML, ebML, DocBook, HRML and RDF, as well as experience designing a custom DTD for a particular project where no standard existed.

8. Using XSLT, how would you extract a specific attribute from an element in an XML document?

Successful candidates should recognize that this one of the most basic applications of XSLT. If they are not able to construct a reply similar to the example below, they should at least be able to identify the components necessary for this operation: xsl:template to match the appropriate XML element, xsl:value-of to select the attribute value and the optional xsl:apply-templates to continue processing the document.

Extract Attributes from XML Data

Example 1.

```
1    <xsl:template match='element-name'>
2            Attribute Value: <xsl:value-of select='attribute'/>
3            <xsl:apply-templates/>
4    </xsl:template>
```

9. When constructing an XML DTD, how do you create an external entity reference in an attribute value?

Every interview session should have at least one trick question. Although possible when using SGML, XML DTD's don't support defining external entity references in attribute values. It's not as important that the candidate catches this somewhat obscure point as responds to the question in a logical fashion.

10. How would you build a search engine for large volumes of XML data?

The way candidates answer this question provides considerable insight to their view of XML data. For those who view XML primarily as a way to denote structure for text files, a common answer is to build a full-text search and handle the data similarly to the way internet portals handle HTML pages. Others consider XML as a standard way of transferring fielded data between disparate systems. These candidates often describe some scheme of importing XML into a relational or object database and relying on the database's engine for searching. Lastly, candidates that have worked with vendors specializing in this area often cite that the best way the handle this situation is to use a third party software package optimized for XML data.

Obviously, some important areas of XML technologies were not included in this list (namespaces, XPointer, XLink, etc.) and should be added to the interviewer's set of questions if applicable to the particular position that the candidate is applying for.However, these questions in conjunction with others to assess soft skills (communication skills, ability to work on teams, leadership ability, etc.) will help determine how well candidates understand the fundamental principles of XML.

Top 10 Interview Questions for DITA Developers