XML Interview Questions with Answers:

11. 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, to allow the candidate to explain the rationale

for choosing XML by citing a few real world examples. For instance,

one appropriate answer is that XML allows content management systems

to store documents independently of their format, which thereby

reduces 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.

12. 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 an application independent

mechanism to access, parse, or update XML data. In simple terms

it is a hierarchical model that allows developers to manipulate

XML documents easily 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 an event-based interface like

SAX.

13. 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 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.

14. Why not just carry on extending HTML?

HTML was already overburdened with dozens of interesting but incompatible

inventions from different manufacturers, because it provides only

one way of describing your information.

XML allows groups of people or organizations to question C.13, create

their own customized markup applications for exchanging information

in their domain (music, chemistry, electronics, hill-walking, finance,

surfing, petroleum geology, linguistics, cooking, knitting, stellar

cartography, history, engineering, rabbit-keeping, question C.19,

mathematics, genealogy, etc).

HTML is now well beyond the limit of its usefulness as a way of

describing information, and while it will continue to play an important

role for the content it currently represents, many new applications

require a more robust and flexible infrastructure.

15. Why should I use XML?

Here are a few reasons for using XML (in no particular order).

Not all of these will apply to your own requirements, and you may

have additional reasons not mentioned here (if so, please let the

editor of the FAQ know!).

\* XML can be used to describe and identify information accurately

and unambiguously, in a way that computers can be programmed to

‘understand’ (well, at least manipulate as if they could

understand).

\* XML allows documents which are all the same type to be created

consistently and without structural errors, because it provides

a standardized way of describing, controlling, or allowing/disallowing

particular types of document structure. [Note that this has absolutely

nothing whatever to do with formatting, appearance, or the actual

text content of your documents, only the structure of them.]

\* XML provides a robust and durable format for information storage

and transmission. Robust because it is based on a proven standard,

and can thus be tested and verified; durable because it uses plain-text

file formats which will outlast proprietary binary ones.

\* XML provides a common syntax for messaging systems for the exchange

of information between applications. Previously, each messaging

system had its own format and all were different, which made inter-system

messaging unnecessarily messy, complex, and expensive. If everyone

uses the same syntax it makes writing these systems much faster

and more reliable.

\* XML is free. Not just free of charge (free as in beer) but free

of legal encumbrances (free as in speech). It doesn’t belong to

anyone, so it can’t be hijacked or pirated. And you don’t have to

pay a fee to use it (you can of course choose to use commercial

software to deal with it, for lots of good reasons, but you don’t

pay for XML itself).

\* XML information can be manipulated programmatically (under machine

control), so XML documents can be pieced together from disparate

sources, or taken apart and re-used in different ways. They can

be converted into almost any other format with no loss of information.

\* XML lets you separate form from content. Your XML file contains

your document information (text, data) and identifies its structure:

your formatting and other processing needs are identified separately

in a style sheet or processing system. The two are combined at output

time to apply the required formatting to the text or data identified

by its structure (location, position, rank, order, or whatever).

16. Can you walk us 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.

17. Give some examples of XML DTDs 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 DTDs 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 DTDs such as FpML, DocBook, HRML, and RDF, as well as experience

designing a custom DTD for a particular project where no standard

existed.

18. Using XSLT, how would you extract a specific attribute

from an element in an XML document?

Successful candidates should recognize this as 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.

&lt;xsl:template match=”element-name”&gt;

Attribute Value:

&lt;xsl:value-of select=”@attribute”/&gt;

&lt;xsl:apply-templates/&gt;

&lt;/xsl:template&gt;

19. 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 DTDs don’t support defining

external entity references in attribute values. It’s more important

for the candidate to respond to this question in a logical way than

than the candidate know the somewhat obscure answer.

20. How would you build a search engine for large volumes

of XML data?

The way candidates answer this question may provide insight into

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 structured 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 say that the best way the handle this situation

is to use a third party software package optimized for XML data.