Introduction to XML

Outline

- What is XML?
- Syntax of XML Document
- DTD (Document Type Definition)
- XML Schema
- XML Query Language
- XML Databases
- Oracle JDBC

Introduction to XML

- XML stands for E**X**tensible **M**arkup **L**anguage
- XML was designed to describe data.
- XML tags are not predefined unlike HTML
- XML DTD and XML Schema define rules to describe data
- XML example of semi structured data

XML DTD

- A DTD is a set of rules that allow us to specify our own set of elements and attributes.

- DTD is grammar to indicate what tags are legal in XML documents. c

- XML Document is valid if it has an attached DTD and document is structured according to rules defined in DTD.

DTD Example

```
<BOOKLIST>
<BOOK GENRE = “Science” FORMAT = “Hardcover”>
    <AUTHOR>
      <FIRSTNAME> RICHRD </FIRSTNAME>
   <LASTNAME> KARTER </LASTNAME>
   </AUTHOR>
   </BOOK>
   </BOOKS>
```

```
<!DOCTYPE BOOKLIST[
<!ELEMENT BOOKLIST(BOOK)*>
    <!ELEMENT BOOK(AUTHOR)>
<!ELEMENT AUTHOR(FIRSTNAME,LASTNAME)>
<!ELEMENT FIRSTNAME(#PCDATA)>
<!ELEMENT>LASTNAME(#PCDATA)>
<!ATTLIST BOOK GENRE (Science|Fiction)#REQUIRED>
<!ATTLIST BOOK FORMAT (Paperback|Hardcover) “PaperBack”>]>
```

Xml Document And Corresponding DTD

When to Use a DTD?

- With a DTD, independent groups of people can agree to use a standard DTD for interchanging data.
- With a DTD, you can verify that the data you receive from the outside world is valid.
- You can also use a DTD to verify your own data.

XML Schema

- Serves same purpose as database schema
- Schemas are written in XML
- Set of pre-defined simple types (such as string, integer)
- Allows creation of user-defined complex types
- An XML Schema describes the structure of an XML document, just like a DTD.

XML Schema

- **RDBMS Schema (s_id integer, s_name string, s_status string)**
- **XMLSchema**

```
<Students>
<Student id=“p1”>
<Name>Allan</Name>
<Age>62</Age>
<Email>allan@abc.com
</Email>
</Student>
</Students>
```

```
<xs:schema>
<xs:complexType name = “StudnetType”>
<xs:attribute name=“id” type=“xs:string” />
<xs:element name=“Name” type=“xs:string />
<xs:element name=“Age” type=“xs:integer” />
<xs:element name=“Email” type=“xs:string” />
</xs:complexType>
<xs:element name=“Student”
    type=“StudentType” />
</xs:schema>
```

XML Document and Schema

XML DOM

- The Document Object Model (DOM) defines a standard for accessing and manipulating documents
- The **XML DOM** defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure.
- The XML DOM is a standard for how to get, change, add, and delete XML elements.

XML Parser

- The XML DOM (Document Object Model) defines the properties and methods for accessing and editing XML.
- However, before an XML document can be accessed, it must be loaded into an XML DOM object.
- All modern browsers have a built-in XML parser that can convert text into an XML DOM object.

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>

</bookstore>
```

```
txt = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
```

```
<html>
<body>

<p id="demo"></p>

<script>
var text, parser, xmlDoc;

text = "<bookstore><book>" +
"<title>Everyday Italian</title>" +
"<author>Giada De Laurentiis</author>" +
"<year>2005</year>" +
"</book></bookstore>";

parser = new DOMParser();
xmlDoc = parser.parseFromString(text,"text/xml");

document.getElementById("demo").innerHTML =
xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
</script>

</body> </html>
```

- A text string is defined:

```
text = "<bookstore><book>" +
"<title>Everyday Italian</title>" +
"<author>Giada De Laurentiis</author>" +
"<year>2005</year>" +
"</book></bookstore>";
```

- An XML DOM parser is created

```
parser = new DOMParser();
```

- The parser creates a new XML DOM object using the text string

```
xmlDoc = parser.parseFromString(text,"text/xml");
```