Java DOM Parser - Query XML Document

Java DOM parser is an API in java to parse and query XML documents. Using Java DOM parser, we can query large XML documents to get insights about our data. Manually, it is not easy to check the entire XML document to obtain related information. We can query XML elements by their tag name using getElementsByTagName() method. To query based on attribute value, getAttribute() method can be used.

Query XML Using Java DOM Parser

We can query any XML document in Java using DOM parser through following steps −

- **Step 1:** Creating a DocumentBuilder Object
- **Step 2:** Reading the XML
- **Step 3:** Parsing the XML Document
- **Step 4:** Querying the XML Document

Refer this page for first three steps.

Step4: Querying the XML Document

After following the first three steps we can query the XML document according to our business requirements using DOM methods.

Querying Elements by TagName

We can query the XML elements by their tag name by using the method, **getElementsByTagName('carname')** on the Document interface. This method takes tag name in the form of a String and returns the list of nodes with the same tag name as a NodeList.

The **getTextContent()** method of Node interface gives the text content inside the node in the form of a String.

cars.xml

The cars.xml file has seven **<carname>** elements inside the root element, **<cars>**.

```
<?xml version = "1.0"?>
<cars>
      <carname company="Ferrari" >Ferrari 101</carname>
      <carname company="Lamborghini">Lamborghini 001</carname>
      <carname company="Lamborghini">Lamborghini 002</carname>
      <carname company="Lamborghini">Lamborghini 003</carname>
      <carname company="Bentley">Bentley 1</carname>
      <carname company="Bentley">Bentley 2</carname>
      <carname company="Bentley">Bentley 3</carname>
</cars>
```

Query XML Elements

In the following program, we got all the nodes into a nodeList and then iterated each node to get the text content and checked if that equals to "Bentley 2". If it is found, we are printing on the console that it is found or else we are printing not found.

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import java.io.File;

public class QueryXMLDemo {

   public static void main(String argv[]) {

      try {

         //Creating a DocumentBuilder Object
         DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
         DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

         //Reading the XML file
         File inputFile = new File("cars.xml");

         //Parsing the XML Document
         Document doc = dBuilder.parse(inputFile);

         //checking "Bentley 2" car
         int flag=0;
         NodeList nList = doc.getElementsByTagName("carname");
         for(int i=0;i<nList.getLength();i++) {
            if(nList.item(i).getTextContent().equals("Bentley 2")) {
                System.out.println("Bentley 2 car is "+"found");
                flag=1;
                break;
            }
         }
         if(flag==0) {
            System.out.println("Bentley 2 car is "+"not found");
         }
      } catch (Exception e) {e.printStackTrace();}
   }
}
```

Output

Since, "Bentley 2" car is present, it displays that it is found.

```
Bentley 2 car is found
```

Querying Elements by Attributes

We can query the elements by their attributes using the method **getAttribute("Attribute_name")** of Element interface. This method takes attribute name as an argument and returns value of the attribute.

Example

In the following program, we have parsed **cars.xml** file and incremented the count variable every time "Bentley" is found.

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Element;
import java.io.File;

public class QueryXMLAttributes {

   public static void main(String argv[]) {
      try {

         //Creating a DocumentBuilder Object
         DocumentBuilderFactory dbFactory = 
DocumentBuilderFactory.newInstance();
         DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

         //Reading the XML file
         File inputFile = new File("cars.xml");

         //Parsing the XML Document
         Document doc = dBuilder.parse(inputFile);

         //counting Bentley cars
         int count=0;
         NodeList nList = doc.getElementsByTagName("carname");
         for(int i=0;i<nList.getLength();i++) {
            Element ele = (Element) nList.item(i);
            if(ele.getAttribute("company").equals("Bentley")) {
                count++;
            }
         }
         System.out.println("No of Bentley cars: "+ count);
      } catch (Exception e) {e.printStackTrace();}
   }
}
```

Output

The above program counts the number of Bentley cars and displays on the console.

```
No of Bentley cars: 3
```