Java DOM4J Parser - Parse XML Document

Java DOM4J parser is an API in Java to parse XML documents. It creates DOM4J document from a built-in SAX parser or a DOM parser. After getting the document, we can retrieve the information of elements and attributes using the built-in methods of DOM4J Document and Element interfaces.

In this chapter, we have used getRootElement() to extract root element of the document and the elements() method to get all of its child elements.

Parse XML Using Java DOM4J Parser

Following are the steps used while parsing a document using Java DOM4J Parser –

- **Step 1:** Creating SAXReader object
- **Step 2:** Reading the XML file
- **Step 3:** Parsing the XML
- **Step 4:** Extracting the root
- **Step 5:** Retrieving Elements

Step 1: Creating SAXReader object

The SAXReader class is used to create a DOM4J document from an XML file or stream. It has its own built-in SAX parser to parse the file. We create a SAXReader object as follows –

```
SAXReader reader = new SAXReader();
```

Step 2: Reading the XML file

To read XML content as a string, we can use StringBuilder class and later convert it into a ByteStream to create XML document. If XML content is available in the form of a file, we can read it using File class of java.io as follows –

```
File inputFile = new File("src/input.txt");
```

Step 3: Parsing the XML

To parse the XML file, we have created SAXReader object in step 1. Using the read() method of SAXReader, we create DOM4J document by passing the file we read in step 2 as an argument as follows –

```
Document document = reader.read(inputFile);
```

Step 4: Extracting the Root

The root element needs to be extracted from DOM4J document to obtain any information of elements. Using the getRootElement() method of Document interface we obtain the root element as follows –

```
Element RootElement = document.getRootElement();
```

Step 5: Retrieving Elements

After we have followed the first four steps, we now have the root element to obtain the information of its child elements. Now, we are going to perform some tasks such as retrieving the root, retrieving attributes and retrieving element text of an XML document with examples.

Retrieving the Root

The **getRootElement()** method of Document interface returns the root element of the document in the form of an Element object. To get the name of the element, we use **getName()** method of Element interface. It returns the name of the element in the form of a String.

Example

The **studentData.xml** file we need to parse is as follows –

```
<?xml version = "1.0"?>
<class>
   <student rollno = "393">
      <firstname>dinkar</firstname>
      <lastname>kad</lastname>
      <nickname>dinkar</nickname>
      <marks>85</marks>
   </student>
```

```
   <student rollno = "493">
      <firstname>Vaneet</firstname>
      <lastname>Gupta</lastname>
      <nickname>vinni</nickname>
      <marks>95</marks>
   </student>

   <student rollno = "593">
      <firstname>jasvir</firstname>
      <lastname>singn</lastname>
      <nickname>jazz</nickname>
      <marks>90</marks>
   </student>
</class>
```

The **RetrieveRoot.java** program reads the above studentData.xml file using a SAXReader and obtains a DOM4J document. After getting the document, we use getRootElement() method to extract the root.

```
import java.io.File;
import org.dom4j.Document;
import org.dom4j.Element;
import org.dom4j.io.SAXReader;

public class RetrieveRoot {
   public static void main(String[] args) {
      try {

         //Creating SAXReader
         SAXReader reader = new SAXReader();

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

         //Parsing the XML
         Document document = reader.read(inputFile);

         //Extracting the root
         Element RootElement = document.getRootElement();

         //Printing the Root Element Name
         System.out.println("Root element Name :" + RootElement.getName());
```

```
      } catch(Exception e) {
         e.printStackTrace();
      }
   }
}
```

Output

The root element name, 'class' is displayed on the output screen.

```
Root element Name :class
```

Learn **Java** in-depth with real-world projects through our **Java certification course**. Enroll and become a certified expert to boost your career.

Retrieving Attributes

The **attributeValue()** method of Element interface retrieves the value of the specified attribute in the form of a String. If there is no such attribute for that element, it returns null. If there is no value specified for the attribute, it returns an empty string.

Example

The following **RetrieveAttributes.java** program uses the attributeValue() method and retrieves all the roll numbers of student elements.

```
import java.io.File;
import java.util.List;
import org.dom4j.Document;
import org.dom4j.Element;
import org.dom4j.io.SAXReader;

public class RetrieveAttributes {
   public static void main(String[] args) {
      try {

         //Creating SAXReader
         SAXReader reader = new SAXReader();

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

```
         //Parsing the XML
         Document document = reader.read(inputFile);

         //Extracting the root
         Element RootElement = document.getRootElement();

         //Iterating over the List
         List<Element> elements = RootElement.elements();
         System.out.println("Student Roll numbers - ");
         for (Element ele : elements) {
            System.out.println(ele.attributeValue("rollno") );
         }
      } catch(Exception e) {
         e.printStackTrace();
      }
   }
}
```

All the student roll numbers are displayed on the output screen.

Output

```
Student Roll numbers -
393
493
593
```

Retrieving Element Text

The elements() method of Element interface returns the list of Elements contained in it. The **elementText()** method of Element interface returns the text content of the element in the form of a string.

```
import java.io.File;
import java.util.List;
import org.dom4j.Document;
import org.dom4j.DocumentException;
import org.dom4j.Element;
import org.dom4j.io.SAXReader;

public class DemoParse {
```

```
   public static void main(String[] args) {
      try {

         //Creating SAXReader
         SAXReader reader = new SAXReader();

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

         //Parsing the XML
         Document document = reader.read(inputFile);

         //Extracting the root
         Element RootElement = document.getRootElement();
         System.out.println("Root Element: " + RootElement.getName());
         List<Element> elements = RootElement.elements();
         System.out.println("--------------------------------");

         //Iterating over the List
         for (Element ele : elements) {
            System.out.println("\nCurrent Element :"
               + ele.getName());
            System.out.println("Student roll no : "
               + ele.attributeValue("rollno") );
            System.out.println("First Name : "
               + ele.elementText("firstname"));
            System.out.println("Last Name : "
               + ele.elementText("lastname"));
            System.out.println("First Name : "
               + ele.elementText("nickname"));
            System.out.println("Marks : "
               + ele.elementText("marks"));
         }
      } catch (DocumentException e) {
         e.printStackTrace();
      }
   }
}
```

Output

All the information of students is displayed on the output screen.

```
Root Element: class
---------------------------------

Current Element :student
Student roll no : 393
First Name : dinkar
Last Name : kad
First Name : dinkar
Marks : 85

Current Element :student
Student roll no : 493
First Name : Vaneet
Last Name : Gupta
First Name : vinni
Marks : 95

Current Element :student
Student roll no : 593
First Name : jasvir
Last Name : singn
First Name : jazz
Marks : 90
```