Java DOM4J Parser - Modify XML Document

Java DOM4J API provides methods to modify XML documents. We might come across situations where we need to update few details in a large XML document. For example, an e-commerce website has many products whose price may vary on a daily or monthly basis. In such cases, modifying the already existing file makes the job simpler than to create the XML documents again. In this chapter, we are going to look at some examples to learn how to modify existing XML documents using DOM4J API.

Modify XML Using Java DOM4J Parser

Following are the steps to modify XML documents 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:** Modifying the elements
- **Step 6:** Creating a FileOutputStream
- **Step 7:** Writing the updated XML document into file
- **Step 8:** Printing the XML document

Refer Parse XML Document chapter of this section for the first four steps.

Step 5: Modifying the elements

After extracting the root element in step 4, we can get any of its child elements by using the elements() method. We can modify already existing elements by editing their text content, changing attribute values, adding new attributes etc.

Refer Create XML Document chapter of this section for the last three steps.

Modify Text Content

To modify text content of an element, we can use the method **setText()** to set the text content. It takes a string as an argument and updates the old text content with the new text content. This method is available in both Node and Element interfaces.

Example

In the following **studentData.xml** file, We need to update marks from 80 to 64 for the student with roll number 493.

```
<?xml version="1.0" encoding="UTF-8"?>
<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>80</marks>
   </student>

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

The following **ModifyTextContent.java** program reads the above studentData.xml file using SAXReader. Using elementIterator("student") method, we got all the student elements in an Iterator. We iterate all the elements to find the student with roll number 493 using attributeValue() method and updates the text content of marks element.

```
import java.io.File;
import java.io.FileOutputStream;
import java.util.Iterator;
import org.dom4j.Document;
import org.dom4j.Element;
import org.dom4j.io.OutputFormat;
import org.dom4j.io.SAXReader;
import org.dom4j.io.XMLWriter;

public class ModifyTextContent {
```

```
    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();

          //Modifying the elements
          Iterator<Element> students = RootElement.elementIterator("student");
          while(students.hasNext()) {
             Element student = students.next();
             if(student.attributeValue("rollno").equals("493")) {
                Element marks = student.element("marks");
                marks.setText("64");
             }
          }
          //Creating a FileOutputStream
          FileOutputStream newFile = new FileOutputStream("studentData.xml");

          //Writing the updated XML document into file
          XMLWriter writer = new XMLWriter(newFile);
          writer.write( document );

          //Printing the XML document
          OutputFormat format = OutputFormat.createPrettyPrint();
          XMLWriter consoleWriter = new XMLWriter( System.out, format );
          consoleWriter.write( document );

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

Output

Marks for the student with roll number 493 is updated from 80 to 64.

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

<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>64</marks>
  </student>
  <student rollno="593">
    <firstname>jasvir</firstname>
    <lastname>singn</lastname>
    <nickname>jazz</nickname>
    <marks>90</marks>
  </student>
</class>
```


Add New Elements

The **addElement()** method adds new elements to an existing XML document at the end of all the already existing child elements of that current element. We need to pass the tag name of the element as an argument. Similarly, the addText() method adds text content to the element. The **addAttribute()** adds new attribute to the current element. We need to pass attribute name and attribute value as arguments.

The following **AddNewElements.java** program reads the studentData.xml file we have used in the previous example and uses the above methods to add information of a new student.

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

public class AddNewElements {
   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();

         //Modifying the elements
         Element student =
RootElement.addElement("student").addAttribute("rollno", "693");

         student.addElement("firstname").addText("John");
         student.addElement("lastname").addText("Daniel");
         student.addElement("nickname").addText("Johny");
         student.addElement("marks").addText("78");

         //Creating a FileOutputStream
         FileOutputStream newFile = new FileOutputStream("studentData.xml");

         //Writing the updated XML document into file
         XMLWriter writer = new XMLWriter(newFile);
         writer.write( document );

         //Printing the XML document
         OutputFormat format = OutputFormat.createPrettyPrint();
```

```
            XMLWriter consoleWriter = new XMLWriter( System.out, format );
            consoleWriter.write( document );

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

Output

The file content after adding new student information is as follows –

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

<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>64</marks>
  </student>
  <student rollno="593">
    <firstname>jasvir</firstname>
    <lastname>singn</lastname>
    <nickname>jazz</nickname>
    <marks>90</marks>
  </student>
  <student rollno="693">
    <firstname>John</firstname>
    <lastname>Daniel</lastname>
    <nickname>Johny</nickname>
    <marks>78</marks>
```

```
   </student>
</class>
```