

JAVA SAX PARSER - MODIFY XML DOCUMENT

http://www.tutorialspoint.com/java_xml/java_sax_modify_document.htm

Copyright © tutorialspoint.com

Demo Example

Here is the input xml file we need to Modify by appending <Result>Pass<Result/> at the end of </marks> tag

```
<?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>
```

SAXModifyDemo.java

```
package com.tutorialspoint.xml;

import java.io.*;
import org.xml.sax.*;
import javax.xml.parsers.*;
import org.xml.sax.helpers.DefaultHandler;

public class SAXModifyDemo extends DefaultHandler {
    static String displayText[] = new String[1000];
    static int numberLines = 0;
    static String indentation = "";

    public static void main(String args[]) {
        try {
            File inputFile = new File("input.txt");
            SAXParserFactory factory =
                SAXParserFactory.newInstance();
            SAXModifyDemo obj = new SAXModifyDemo();
            obj.childLoop(inputFile);
            FileWriter filewriter = new FileWriter("newfile.xml");
            for(int loopIndex = 0; loopIndex < numberLines; loopIndex++){
                filewriter.write(displayText[loopIndex].toCharArray());
                filewriter.write('\n');
                System.out.println(displayText[loopIndex].toString());
            }
            filewriter.close();
        }
        catch (Exception e) {
            e.printStackTrace(System.err);
        }
    }
}
```

```

public void childLoop(File input){
    DefaultHandler handler = this;
    SAXParserFactory factory = SAXParserFactory.newInstance();
    try {
        SAXParser saxParser = factory.newSAXParser();
        saxParser.parse(input, handler);
    } catch (Throwable t) {}
}

public void startDocument() {
    displayText[numberLines] = indentation;
    displayText[numberLines] += "<?xml version=\"1.0\" encoding=\"" +
        "UTF-8\" + "\""?">";
    numberLines++;
}

public void processingInstruction(String target,
    String data) {
    displayText[numberLines] = indentation;
    displayText[numberLines] += "<?";
    displayText[numberLines] += target;
    if (data != null && data.length() > 0) {
        displayText[numberLines] += ' ';
        displayText[numberLines] += data;
    }
    displayText[numberLines] += ">";
    numberLines++;
}

public void startElement(String uri, String localName,
    String qualifiedName, Attributes attributes) {
    displayText[numberLines] = indentation;

    indentation += "    ";

    displayText[numberLines] += '<';
    displayText[numberLines] += qualifiedName;
    if (attributes != null) {
        int numberAttributes = attributes.getLength();
        for (int loopIndex = 0; loopIndex < numberAttributes;
            loopIndex++){
            displayText[numberLines] += ' ';
            displayText[numberLines] += attributes.getQName(loopIndex);
            displayText[numberLines] += "=\"";
            displayText[numberLines] += attributes.getValue(loopIndex);
            displayText[numberLines] += "\"";
        }
    }
    displayText[numberLines] += '>';
    numberLines++;
}

public void characters(char characters[],
    int start, int length) {
    String characterData = (new String(characters, start, length)).trim();
    if (characterData.indexOf("\n") < 0 && characterData.length() > 0) {
        displayText[numberLines] = indentation;
        displayText[numberLines] += characterData;
        numberLines++;
    }
}

public void endElement(String uri, String localName,
    String qualifiedName) {
    indentation = indentation.substring(0, indentation.length() - 4) ;
    displayText[numberLines] = indentation;
    displayText[numberLines] += "</";
    displayText[numberLines] += qualifiedName;
}

```

```

displayText[numberLines] += '>';
numberLines++;

if (qualifiedName.equals("marks")) {
    startElement("", "Result", "Result", null);
    characters("Pass".toCharArray(), 0, "Pass".length());
    endElement("", "Result", "Result");
}
}
}

```

This would produce the following result:

```

<?xml version="1.0" encoding="UTF-8"?>
<class>
  <student rollno="393">
    <firstname>
      dinkar
    </firstname>
    <lastname>
      kad
    </lastname>
    <nickname>
      dinkar
    </nickname>
    <marks>
      85
    </marks>
    <Result>
      Pass
    </Result>
  </student>
  <student rollno="493">
    <firstname>
      Vaneet
    </firstname>
    <lastname>
      Gupta
    </lastname>
    <nickname>
      vinni
    </nickname>
    <marks>
      95
    </marks>
    <Result>
      Pass
    </Result>
  </student>
  <student rollno="593">
    <firstname>
      jasvir
    </firstname>
    <lastname>
      singh
    </lastname>
    <nickname>
      jazz
    </nickname>
    <marks>
      90
    </marks>
    <Result>
      Pass
    </Result>
  </student>
</class>

```