

JAVA XPATH PARSER - PARSE XML DOCUMENT

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

Copyright © tutorialspoint.com

Steps to Using XPath

Following are the steps used while parsing a document using XPath Parser.

- Import XML-related packages.
- Create a DocumentBuilder
- Create a Document from a file or stream
- Create an XPath object and an XPath path expression
- Compile the XPath expression using XPath.compile and get a list of nodes by evaluating the compiled expression via XPath.evaluate
- Iterate over the list of nodes.
- Examine attributes
- Examine sub-elements

Import XML-related packages

```
import org.w3c.dom.*;
import org.xml.sax.*;
import javax.xml.parsers.*;
import javax.xml.xpath.*;
import java.io.*;
```

Create a DocumentBuilder

```
DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
```

Create a Document from a file or stream

```
StringBuilder xmlStringBuilder = new StringBuilder();
xmlStringBuilder.append("<?xml version='1.0'?> <class> </class>");
ByteArrayInputStream input = new ByteArrayInputStream(
    xmlStringBuilder.toString().getBytes("UTF-8"));
Document doc = builder.parse(input);
```

Build XPath

```
XPath xPath = XPathFactory.newInstance().newXPath();
```

Prepare Path expression and evaluate it

```
String expression = "/class/student";
NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);
```

Iterate over NodeList

```
for (int i = 0; i < nodeList.getLength(); i++) {
    Node nNode = nodeList.item(i);
    ...
}
```

Examine attributes

```
//returns specific attribute
getAttribute("attributeName");
//returns a Map (table) of names/values
getAttributes();
```

Examine sub-elements

```
//returns a list of subelements of specified name
getElementsByTagName("subelementName");
//returns a list of all child nodes
getChildNodes();
```

Demo Example

Here is the input text file we need to parse:

```
<?xml version="1.0"?>
<ltclass>
  <ltstudent rollno="393">
    <ltfirstname>dinkar</firstname>
    <ltlastname>kad</lastname>
    <ltnickname>dinkar</nickname>
    <ltmarks>85</marks>
  </student>
  <ltstudent rollno="493">
    <ltfirstname>Vaneet</firstname>
    <ltlastname>Gupta</lastname>
    <ltnickname>vinni</nickname>
    <ltmarks>95</marks>
  </student>
  <ltstudent rollno="593">
    <ltfirstname>jasvir</firstname>
    <ltlastname>singh</lastname>
    <ltnickname>jazz</nickname>
    <ltmarks>90</marks>
  </student>
</class>
```

Demo Example:

XPathParserDemo.java

```
package com.tutorialspoint.xml;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import org.xml.sax.SAXException;

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

```

File inputFile = new File("input.txt");
DocumentBuilderFactory dbFactory
    = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder;

dBuilder = dbFactory.newDocumentBuilder();

Document doc = dBuilder.parse(inputFile);
doc.getDocumentElement().normalize();

XPath xPath = XPathFactory.newInstance().newXPath();

String expression = "/class/student";
NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);
for (int i = 0; i < nodeList.getLength(); i++) {
    Node nNode = nodeList.item(i);
    System.out.println("\nCurrent Element : "
        + nNode.getNodeName());
    if (nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) nNode;
        System.out.println("Student roll no : "
            + eElement.getAttribute("rollno"));
        System.out.println("First Name : "
            + eElement
                .getElementsByTagName("firstname")
                .item(0)
                .getTextContent());
        System.out.println("Last Name : "
            + eElement
                .getElementsByTagName("lastname")
                .item(0)
                .getTextContent());
        System.out.println("Nick Name : "
            + eElement
                .getElementsByTagName("nickname")
                .item(0)
                .getTextContent());
        System.out.println("Marks : "
            + eElement
                .getElementsByTagName("marks")
                .item(0)
                .getTextContent());
    }
}
} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (XPathExpressionException e) {
    e.printStackTrace();
}
}
}

```

This would produce the following result:

```

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

Current Element :student
Student roll no : 493
First Name : Vaneet

```

Last Name : Gupta
Nick Name : vinni
Marks : 95

Current Element :student
Student roll no : 593
First Name : jasvir
Last Name : singh
Nick Name : jazz
Marks : 90

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js