

Chapters

Upcoming Trainings

Selenium Interview Questions

File Handling

Regular Expression

XML

Database

Selenium Installation with Maven

TestNg Installation

LOADRUNNER Job Openings

Mobile App Testing Job Openings

Selenium Openings

QTP/UFT Openings

Contact Us

 info@jaseleniumworld.com

 +91- 9886577710

XML

Share 53

On this page you can find a simple guide to reading and writing XML files. The code examples here give you everything you need to read and write to XML files.

- a. [Read XML file](#)
- b. [Write XML file](#)

a. Read XML file

This Section show you how to read an XML file content

XML File Content(Book.xml)

```
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
      with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
  </book>
</catalog>
```

Code: To read name of Authors(Gambardella, Matthew & Ralls, Kim) from the above XML file(Book.xml)

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

public class ReadXML {
    public static void main(String[] args) throws Exception {

        //The two lines below are just for getting an
        //instance of DocumentBuilder which we use
        //for parsing XML data
        DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();

        //Parsing of Book.xml is done here
        Document doc = builder.parse(new File("C:/Javaseleniumworld/Book.xml"));

        //Here we get the root element of XML and print out
        Element rootElement = doc.getDocumentElement();

        //Here we get a list of all elements named 'child'
        NodeList list = rootElement.getElementsByTagName("author");

        //Traversing all the elements from the list and printing
        //out its data
        for (int i = 0; i < list.getLength(); i++) {

            //Getting one node from the list.
            Node childNode = list.item(i);

            System.out.println("Name of the Author : " + childNode.getTextContent());

        }
    }
}
```

Output:

```
Name of the Author : Gambardella, Matthew
Name of the Author : Ralls, Kim
```

b. Write an XML file

This Section will show you how to write a fresh XML with your defined tags

Code: A XML file with fresh set of tags and value is created

```

import java.io.File;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.Attr;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class WriteXML {

    public static void main(String argv[]) {

        try {

            DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder docBuilder = docFactory.newDocumentBuilder();

            // root elements
            Document doc = docBuilder.newDocument();
            Element rootElement = doc.createElement("Catalog");
            doc.appendChild(rootElement);

            // Book elements
            Element staff = doc.createElement("Book");
            rootElement.appendChild(staff);

            // set attribute to Book element
            Attr attr = doc.createAttribute("id");
            attr.setValue("1");
            staff.setAttributeNode(attr);

            // Author elements
            Element firstname = doc.createElement("author");
            firstname.appendChild(doc.createTextNode("JavaSeleniumWorld.com"));
            staff.appendChild(firstname);

            // title elements
            Element lastname = doc.createElement("title");
            lastname.appendChild(doc.createTextNode("Selenium"));
            staff.appendChild(lastname);

            // genre elements
            Element nickname = doc.createElement("genre");
            nickname.appendChild(doc.createTextNode("Java"));
            staff.appendChild(nickname);

            // price elements
            Element salary = doc.createElement("price");
            salary.appendChild(doc.createTextNode("20"));
            staff.appendChild(salary);

            // write the content into xml file
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);
            StreamResult result = new StreamResult(new File("C:/Javaseleniumworld/file.xml"));

            transformer.transform(source, result);

            System.out.println("File saved!");

        } catch (ParserConfigurationException pce) {
            pce.printStackTrace();
        } catch (TransformerException tfe) {
            tfe.printStackTrace();
        }
    }
}

```

Output

XML file "file.xml" with below mentioned content is created

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Catalog>
  <Book id="1">
    <author>JavaSeleniumWorld.com</author>
    <title>Selenium</title>
    <genre>Java</genre>
    <price>20</price>
  </Book>
</Catalog>

```

\$500 Off Your First Order

Deploy an App w/ SoftLayer® Server in India & Save \$500. Act Quickly!

