1. What is Document Object Model (DOM)

The Document Object Model (DOM) uses nodes to represent the HTML or XML document as a tree structure.

Below is a simple XML document:

<company>

<staff id="1001">

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary currency="USD">100000</salary>

</staff>

</company>

DOM common terms.

* The <company> is the root element.
* The <staff>, <firstname> and all <?> are the element nodes.
* The text node is the value wrapped by the element nodes; for example, <firstname>yong</firstname>, the yong is the text node.
* The attribute is part of the element node; for example, <staff id="1001"> the id is the attribute of the staff element.

*Further Reading*

* Wikipedia – Document Object Model (DOM)
* Mozilla – Document Object Model (DOM)

2. Read or Parse a XML file

This example shows you how to use the Java built-in DOM parser APIs to read or parse an XML file.

2.1 Review below XML file.

/users/mkyong/staff.xml

<?xml version="1.0"?>

<company>

<staff id="1001">

<firstname>yong</firstname>

<lastname>mook kim</lastname>

<nickname>mkyong</nickname>

<salary currency="USD">100000</salary>

</staff>

<staff id="2001">

<firstname>low</firstname>

<lastname>yin fong</lastname>

<nickname>fong fong</nickname>

<salary currency="INR">200000</salary>

</staff>

</company>

2.2 Below is a DOM parser example of parsing or reading the above XML file.

ReadXmlDomParser.java

package com.mkyong.xml.dom;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import java.io.File;

import java.io.IOException;

import java.io.InputStream;

public class ReadXmlDomParser {

private static final String FILENAME = "/users/mkyong/staff.xml";

public static void main(String[] args) {

*// Instantiate the Factory*

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

try {

*// optional, but recommended*

*// process XML securely, avoid attacks like XML External Entities (XXE)*

dbf.setFeature(XMLConstants.FEATURE\_SECURE\_PROCESSING, true);

*// parse XML file*

DocumentBuilder db = dbf.newDocumentBuilder();

Document doc = db.parse(new File(FILENAME));

*// optional, but recommended*

*// http://stackoverflow.com/questions/13786607/normalization-in-dom-parsing-with-java-how-does-it-work*

doc.getDocumentElement().normalize();

System.out.println("Root Element :" + doc.getDocumentElement().getNodeName());

System.out.println("------");

*// get <staff>*

NodeList list = doc.getElementsByTagName("staff");

for (int temp = 0; temp < list.getLength(); temp++) {

Node node = list.item(temp);

if (node.getNodeType() == Node.ELEMENT\_NODE) {

Element element = (Element) node;

*// get staff's attribute*

String id = element.getAttribute("id");

*// get text*

String firstname = element.getElementsByTagName("firstname").item(0).getTextContent();

String lastname = element.getElementsByTagName("lastname").item(0).getTextContent();

String nickname = element.getElementsByTagName("nickname").item(0).getTextContent();

NodeList salaryNodeList = element.getElementsByTagName("salary");

String salary = salaryNodeList.item(0).getTextContent();

*// get salary's attribute*

String currency = salaryNodeList.item(0).getAttributes().getNamedItem("currency").getTextContent();

System.out.println("Current Element :" + node.getNodeName());

System.out.println("Staff Id : " + id);

System.out.println("First Name : " + firstname);

System.out.println("Last Name : " + lastname);

System.out.println("Nick Name : " + nickname);

System.out.printf("Salary [Currency] : %,.2f [%s]%n%n", Float.parseFloat(salary), currency);

}

}

} catch (ParserConfigurationException | SAXException | IOException e) {

e.printStackTrace();

}

}

}

Output

Terminal

Root Element :company

------

Current Element :staff

Staff Id : 1001

First Name : yong

Last Name : mook kim

Nick Name : mkyong

Salary [Currency] : 100,000.00 [USD]

Current Element :staff

Staff Id : 2001

First Name : low

Last Name : yin fong

Nick Name : fong fong

Salary [Currency] : 200,000.00 [INR]

3. Read or Parse XML file (Unicode)

In DOM parser, there is no difference between reading a normal and Unicode XML file.

3.1 Review below XML file containing some Chinese characters (Unicode).

src/main/resources/staff-unicode.xml

<?xml version="1.0"?>

<company>

<staff id="1001">

<firstname>揚</firstname>

<lastname>木金</lastname>

<nickname>mkyong</nickname>

<salary currency="USD">100000</salary>

</staff>

<staff id="2001">

<firstname>low</firstname>

<lastname>yin fong</lastname>

<nickname>fong fong</nickname>

<salary currency="INR">200000</salary>

</staff>

</company>

3.2 The below example parse the above XML file; it loops all the nodes one by one and prints it out.

ReadXmlDomParserLoop.java

package com.mkyong.xml.dom;

import org.w3c.dom.\*;

import org.xml.sax.SAXException;

import javax.xml.XMLConstants;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import java.io.IOException;

import java.io.InputStream;

public class ReadXmlDomParserLoop {

public static void main(String[] args) {

*// Instantiate the Factory*

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

try (InputStream is = readXmlFileIntoInputStream("staff-unicode.xml")) {

*// parse XML file*

DocumentBuilder db = dbf.newDocumentBuilder();

*// read from a project's resources folder*

Document doc = db.parse(is);

System.out.println("Root Element :" + doc.getDocumentElement().getNodeName());

System.out.println("------");

if (doc.hasChildNodes()) {

printNote(doc.getChildNodes());

}

} catch (ParserConfigurationException | SAXException | IOException e) {

e.printStackTrace();

}

}

private static void printNote(NodeList nodeList) {

for (int count = 0; count < nodeList.getLength(); count++) {

Node tempNode = nodeList.item(count);

*// make sure it's element node.*

if (tempNode.getNodeType() == Node.ELEMENT\_NODE) {

*// get node name and value*

System.out.println("\nNode Name =" + tempNode.getNodeName() + " [OPEN]");

System.out.println("Node Value =" + tempNode.getTextContent());

if (tempNode.hasAttributes()) {

*// get attributes names and values*

NamedNodeMap nodeMap = tempNode.getAttributes();

for (int i = 0; i < nodeMap.getLength(); i++) {

Node node = nodeMap.item(i);

System.out.println("attr name : " + node.getNodeName());

System.out.println("attr value : " + node.getNodeValue());

}

}

if (tempNode.hasChildNodes()) {

*// loop again if has child nodes*

printNote(tempNode.getChildNodes());

}

System.out.println("Node Name =" + tempNode.getNodeName() + " [CLOSE]");

}

}

}

*// read file from project resource's folder.*

private static InputStream readXmlFileIntoInputStream(final String fileName) {

return ReadXmlDomParserLoop.class.getClassLoader().getResourceAsStream(fileName);

}

}

Output

Terminal

Root Element :company

------

Node Name =company [OPEN]

Node Value =

揚

木金

mkyong

100000

low

yin fong

fong fong

200000

Node Name =staff [OPEN]

Node Value =

揚

木金

mkyong

100000

attr name : id

attr value : 1001

Node Name =firstname [OPEN]

Node Value =揚

Node Name =firstname [CLOSE]

Node Name =lastname [OPEN]

Node Value =木金

Node Name =lastname [CLOSE]

Node Name =nickname [OPEN]

Node Value =mkyong

Node Name =nickname [CLOSE]

Node Name =salary [OPEN]

Node Value =100000

attr name : currency

attr value : USD

Node Name =salary [CLOSE]

Node Name =staff [CLOSE]

Node Name =staff [OPEN]

Node Value =

low

yin fong

fong fong

200000

attr name : id

attr value : 2001

Node Name =firstname [OPEN]

Node Value =low

Node Name =firstname [CLOSE]

Node Name =lastname [OPEN]

Node Value =yin fong

Node Name =lastname [CLOSE]

Node Name =nickname [OPEN]

Node Value =fong fong

Node Name =nickname [CLOSE]

Node Name =salary [OPEN]

Node Value =200000

attr name : currency

attr value : INR

Node Name =salary [CLOSE]

Node Name =staff [CLOSE]

Node Name =company [CLOSE]

4. Parse Alexa API XML Response

This example shows how to use the DOM parser to parse the XML response from Alexa’s API.

4.1 Send a request to the following Alexa API.

Terminal

https://data.alexa.com/data?cli=10&url=mkyong.com

4.2 The Alexa API will return the following XML response. The Alexa ranking is inside the POPULARITY element, the TEXT attribute.

*<!-- Need more Alexa data? Find our APIs here: https://aws.amazon.com/alexa/ -->*

<ALEXA VER="0.9" URL="mkyong.com/" HOME="0" AID="=" IDN="mkyong.com/">

<SD>

<POPULARITY URL="mkyong.com/" TEXT="20162" SOURCE="panel"/>

<REACH RANK="14430"/>

<RANK DELTA="+947"/>

<COUNTRY CODE="IN" NAME="India" RANK="4951"/>

</SD>

</ALEXA>

4.3 We use a DOM parser to directly select the POPULARITY element and print out the value of the TEXT attribute.

ReadXmlAlexaApi.java

package com.mkyong.xml.dom;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.NodeList;

import javax.xml.XMLConstants;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import java.io.InputStream;

import java.net.URL;

import java.net.URLConnection;

public class ReadXmlAlexaApi {

private static final String ALEXA\_API = "http://data.alexa.com/data?cli=10&url=";

private final DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

public static void main(String[] args) {

ReadXmlAlexaApi obj = new ReadXmlAlexaApi();

int alexaRanking = obj.getAlexaRanking("mkyong.com");

System.out.println("Ranking: " + alexaRanking);

}

public int getAlexaRanking(String domain) {

int result = 0;

String url = ALEXA\_API + domain;

try {

URLConnection conn = new URL(url).openConnection();

try (InputStream is = conn.getInputStream()) {

*// unknown XML better turn on this*

dbf.setFeature(XMLConstants.FEATURE\_SECURE\_PROCESSING, true);

DocumentBuilder dBuilder = dbf.newDocumentBuilder();

Document doc = dBuilder.parse(is);

Element element = doc.getDocumentElement();

*// find this tag "POPULARITY"*

NodeList nodeList = element.getElementsByTagName("POPULARITY");

if (nodeList.getLength() > 0) {

Element elementAttribute = (Element) nodeList.item(0);

String ranking = elementAttribute.getAttribute("TEXT");

if (!"".equals(ranking)) {

result = Integer.parseInt(ranking);

}

}

}

} catch (Exception e) {

e.printStackTrace();

throw new IllegalArgumentException("Invalid request for domain : " + domain);

}

return result;

}

}

The domain mkyong.com ranked 20162.

Terminal

Ranking: 20162

**Note**  
More DOM