DOM4J is an open source, java based library to parse XML document and it is highly flexible, high-performance, and memory-efficient API. It is java optimized, it uses java collection like List and Arrays. It works with DOM, SAX, XPath and XSLT. It can parse large XML document with very low memory footprint.

Environment Setup

In order to use DOM4J parser, you should have dom4j-1.6.1.jar and jaxen.jar in your application's classpath. Download [**dom4j-1.6.1.zip.**](http://sourceforge.net/projects/dom4j/files/latest/download?source=files)

When to use?

You should use a DOM4J parser when:

* You need to know a lot about the structure of a document
* You need to move parts of the document around (you might want to sort certain elements, for example)
* You need to use the information in the document more than once
* You are a java developer and want to leverage java optimized parsing of XML.

What you get?

When you parse an XML document with a DOM4J parser, you get the flexibility to get back a tree structure that contains all of the elements of your document without impacting the memory footprint of the application. The DOM4J provides a variety of utility functions you can use to examine the contents and structure of the document in case document is well structured and its structure is known. DOM4J uses XPath expression to navigate through the XML document.

Advantages

DOM4J gives java developers flexibility and easy maintainablity of xml parsing code. It is light weight and quick API.

DOM4J classes

The DOM4J defines several Java classes. Here are the most common classes:

* **Document** - Represents the entire XML document. A Document object is often referred to as a DOM tree.
* **Element** - Represents an XML element. Element object has methods to manipulate its child elements,its text, attributes and namespaces.
* **Attribute** Represents an attribute of an element. Attribute has method to get and set the value of attribute. It has parent and attribute type.
* **Node** Represents Element, Attribute or ProcessingInstruction

Common DOM4J methods

When you are working with the DOM4J, there are several methods you'll use often:

* **SAXReader.read(xmlSource)()** - Build the DOM4J document from the xml source.
* **Document.getRootElement()** - Get the root element of the XML.
* **Element.node(index)** - Get the XML node at particular index in the element.
* **Element.attributes()** - Get all the attributes of an element.
* **Node.valueOf(@Name)** - Get the value of an attribute with given name of the element.

# Java DOM4J Parser - Parse XML Document

Steps to Using DOM4J

Following are the steps used while parsing a document using DOM4J Parser.

* Import XML-related packages.
* Create a SAXReader
* Create a Document from a file or stream
* Get the required nodes using XPath Expression by calling document.selectNodes()
* Extract the root element
* Iterate over the list of nodes.
* Examine attributes
* Examine sub-elements

**Import XML-related packages**

import java.io.\*;

import java.util.\*;

import org.dom4j.\*;

**Create a DocumentBuilder**

SAXBuilder saxBuilder = new SAXBuilder();

**Create a Document from a file or stream**

File inputFile = new File("input.txt");

SAXBuilder saxBuilder = new SAXBuilder();

Document document = saxBuilder.build(inputFile);

**Extract the root element**

Element classElement = document.getRootElement();

**Examine attributes**

//returns specific attribute

valueOf("@attributeName");

**Examine sub-elements**

//returns first child node

selectSingleNode("subelementName");

Demo Example

**Here is the input xml file we need to parse:**

<?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>

**Demo Example:**

*DOM4JParserDemo.java*

package com.tutorialspoint.xml;

import java.io.File;

import java.util.List;

import org.dom4j.Document;

import org.dom4j.DocumentException;

import org.dom4j.Element;

import org.dom4j.Node;

import org.dom4j.io.SAXReader;

public class DOM4JParserDemo {

public static void main(String[] args) {

try {

File inputFile = new File("input.txt");

SAXReader reader = new SAXReader();

Document document = reader.read( inputFile );

System.out.println("Root element :"

+ document.getRootElement().getName());

Element classElement = document.getRootElement();

List<Node> nodes = document.selectNodes("/class/student" );

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

for (Node node : nodes) {

System.out.println("\nCurrent Element :"

+ node.getName());

System.out.println("Student roll no : "

+ node.valueOf("@rollno") );

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

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

System.out.println("First Name : " + node.selectSingleNode("nickname").getText());

System.out.println("Marks : " + node.selectSingleNode("marks").getText());

}

} catch (DocumentException e) {

e.printStackTrace();

}

}

}

This would produce the following result:

Root element :class

----------------------------

Current Element :student

Student roll no :

First Name : dinkar

Last Name : kad

First Name : dinkar

Marks : 85

Current Element :student

Student roll no :

First Name : Vaneet

Last Name : Gupta

First Name : vinni

Marks : 95

Current Element :student

Student roll no :

First Name : jasvir

Last Name : singn

First Name : jazz

Marks : 90

# Java DOM4J Parser - Query XML Document

## Demo Example

**Here is the input xml file we need to parse:**

<?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>

**Demo Example:**

*DOM4JQueryDemo.java*

package com.tutorialspoint.xml;

import java.io.File;

import java.util.List;

import org.dom4j.Document;

import org.dom4j.DocumentException;

import org.dom4j.Element;

import org.dom4j.Node;

import org.dom4j.io.SAXReader;

public class DOM4JQueryDemo {

public static void main(String[] args) {

try {

File inputFile = new File("input.txt");

SAXReader reader = new SAXReader();

Document document = reader.read( inputFile );

System.out.println("Root element :"

+ document.getRootElement().getName());

Element classElement = document.getRootElement();

List<Node> nodes = document.selectNodes("/class/student[@rollno='493']" );

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

for (Node node : nodes) {

System.out.println("\nCurrent Element :"

+ node.getName());

System.out.println("Student roll no : "

+ node.valueOf("@rollno") );

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

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

System.out.println("First Name : " + node.selectSingleNode("nickname").getText());

System.out.println("Marks : " + node.selectSingleNode("marks").getText());

}

} catch (DocumentException e) {

e.printStackTrace();

}

}

}

This would produce the following result:

Root element :class

----------------------------

Current Element :student

Student roll no : 493

First Name : Vaneet

Last Name : Gupta

First Name : vinni

Marks : 95

# Java DOM4J Parser - Create XML Document

## Demo Example

**Here is the XML we need to create:**

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

<cars>

<supercars company="Ferrari">

<carname type="formula one">Ferrari 101</carname>

<carname type="sports">Ferrari 202</carname>

</supercars>

</cars>

**Demo Example:**

*CreateXmlFileDemo.java*

package com.tutorialspoint.xml;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import org.dom4j.Document;

import org.dom4j.DocumentHelper;

import org.dom4j.Element;

import org.dom4j.io.OutputFormat;

import org.dom4j.io.XMLWriter;

public class DOM4JCreateXMLDemo {

public static void main(String[] args) {

try {

Document document = DocumentHelper.createDocument();

Element root = document.addElement( "cars" );

Element supercarElement= root.addElement("supercars")

.addAttribute("company", "Ferrai");

supercarElement.addElement("carname")

.addAttribute("type", "Ferrari 101")

.addText("Ferrari 101");

supercarElement.addElement("carname")

.addAttribute("type", "sports")

.addText("Ferrari 202");

// Pretty print the document to System.out

OutputFormat format = OutputFormat.createPrettyPrint();

XMLWriter writer;

writer = new XMLWriter( System.out, format );

writer.write( document );

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

This would produce the following result:

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

<cars>

<supercars company="Ferrari">

<carname type="formula one">Ferrari 101</carname>

<carname type="sports">Ferrari 202</carname>

</supercars>

</cars>

# Java DOM4J Parser - Modify XML Document

## Demo Example

**Here is the XML we need to modify:**

<?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>

**Demo Example:**

*DOM4jModifyXMLDemo.java*

package com.tutorialspoint.xml;

import java.io.File;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import java.util.Iterator;

import java.util.List;

import org.dom4j.Document;

import org.dom4j.DocumentException;

import org.dom4j.Element;

import org.dom4j.Node;

import org.dom4j.io.OutputFormat;

import org.dom4j.io.SAXReader;

import org.dom4j.io.XMLWriter;

public class DOM4jModifyXMLDemo {

public static void main(String[] args) {

try {

File inputFile = new File("input.txt");

SAXReader reader = new SAXReader();

Document document = reader.read( inputFile );

Element classElement = document.getRootElement();

List<Node> nodes = document.selectNodes("/class/student[@rollno='493']" );

for (Node node : nodes) {

Element element = (Element)node;

Iterator<Element> iterator=element.elementIterator("marks");

while(iterator.hasNext()){

Element marksElement=(Element)iterator.next();

marksElement.setText("80");

}

}

// Pretty print the document to System.out

OutputFormat format = OutputFormat.createPrettyPrint();

XMLWriter writer;

writer = new XMLWriter( System.out, format );

writer.write( document );

} catch (DocumentException e) {

e.printStackTrace();

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

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>

</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>