Back End Development 1

Sesi 9

Parsing XML Data - SAX & DOM

Introduction

XML parsing adalah parsing yang dirancang untuk membaca XML dan menciptakan cara bagi program untuk menggunakan XML.

Apa itu XML ?

e**X**tensible **M**arkup **L**anguage digunakan untuk menyimpan dan mengantarkan data.

Nah pada sesi kali ini kita akan belajar basic awal bagaimana java bisa melakukan parsing data dalam bentuk XML, yang notabene dalam pekerjaan sehari-hari akan selalu berurusan dengan API.

Untuk Pembahasan API akan dibahas detail pada sesi 16 dan seterusnya.

Contoh Data XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<users>
    <user id="1">
        <id>26</id>
        <name>Bryant</name>
        <gender>Male</gender>
        <role>Java Developer</role>
    </user>
    <user id="2">
        <id>35</id>
        <name>Lisa</name>
        <gender>Female</gender>
        <role>CEO</role>
    </user>
    <user id="3">
        <id>40</id>
        <name>Tom</name>
        <gender>Male</gender>
        <role>Manager</role>
    </user>
    <user id="4">
        <id>25</id>
        <name>Meghna</name>
        <gender>Female</gender>
        <role>Manager</role>
    </user>
</users>
```

Parsing XML Data - SAX & DOM - Sesi 9

Introduction

Teknik Parsing Data di Java :
1. SAX (Simple Api XML)
2. DOM (Document Object Model)

menggunakan SAX

1. Cara Baca File Xml di Java dengan file baru ***User.java***

```java
public class User {

    int id;
    private String Name;
    private String Gender;
    private String role;

    public User() {
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return Name;
    }

    public void setName(String Name) {
        this.Name = Name;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender= gender;
    }

    public String getRole() {
        return role;
    }

    public void setRole(String role) {
        this.role = Role;
    }

    @Override
    public String toString() {

        StringBuilder builder = new StringBuilder();
        builder.append("User{").append("id=").append(id)
                .append(", Name=").append(Name)
                .append(", gender=").append(gender)
                .append(", role=").append(role).append("}");

        return builder.toString();
    }
}
```

Selanjutnya kita implementasi user bean tujuannya untuk hold-data dari file XML,

MyRunner.java

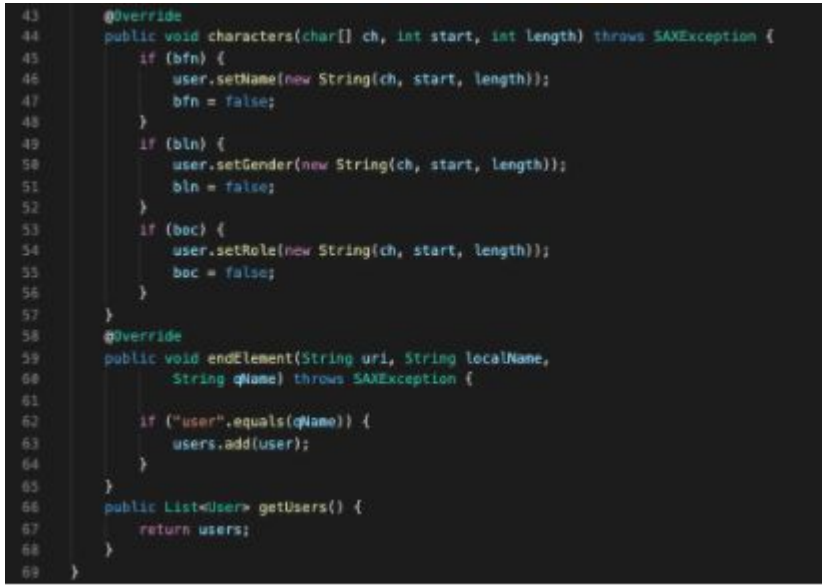
```java
import java.io.File;
import java.io.IOException;
import java.nio.file.Paths;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.SAXException;

public class MyRunner {

    private SAXParser createSaxParser() {

        SAXParser saxParser = null;

        try {

            SAXParserFactory factory = SAXParserFactory.newInstance();
            saxParser = factory.newSAXParser();

            return saxParser;
        } catch (ParserConfigurationException | SAXException ex) {

            Logger lgr = Logger.getLogger(MyRunner.class.getName());
            lgr.log(Level.SEVERE, ex.getMessage(), ex);
        }

        return saxParser;
    }

    public List<User> parseUsers() {

        MyHandler handler = new MyHandler();
        String fileName = "src/main/resources/users.xml";
        File xmlDocument = Paths.get(fileName).toFile();

        try {

            SAXParser parser = createSaxParser();
            parser.parse(xmlDocument, handler);

        } catch (SAXException | IOException ex) {

            Logger lgr = Logger.getLogger(MyRunner.class.getName());
            lgr.log(Level.SEVERE, ex.getMessage(), ex);
        }

        return handler.getUsers();
    }
}
```

Nah Kita telah membuat dokument parse dengan method parse() ,

Selanjutnya Paramenter kedua dari method adalah Objek Handler, dimana terdiri dari event handlers.

Buat 1 File Baru : ***MyHandler.java***

```java
import java.util.ArrayList;
import java.util.List;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class MyHandler extends DefaultHandler {
    private List<User> users = new ArrayList<>();
    private User user;

    private boolean bfn = false;
    private boolean bln = false;
    private boolean boc = false;

    @Override
    public void startElement(String uri, String localName,
            String qName, Attributes attributes) throws SAXException {

        if ("user".equals(qName)) {

            user = new User();

            int id = Integer.valueOf(attributes.getValue("id"));
            user.setId(id);
        }
        switch (qName) {

            case "name":
                bfn = true;
                break;

            case "gender":
                bln = true;
                break;

            case "role":
                boc = true;
                break;
        }
    }
```



```
43      @Override
44      public void characters(char[] ch, int start, int length) throws SAXException {
45          if (bfn) {
46              user.setName(new String(ch, start, length));
47              bfn = false;
48          }
49          if (bln) {
50              user.setGender(new String(ch, start, length));
51              bln = false;
52          }
53          if (boc) {
54              user.setRole(new String(ch, start, length));
55              boc = false;
56          }
57      }
58      @Override
59      public void endElement(String uri, String localName,
60              String qName) throws SAXException {
61
62          if ("user".equals(qName)) {
63              users.add(user);
64          }
65      }
66      public List<User> getUsers() {
67          return users;
68      }
69  }
```

Terakhir dari element <user>, kita tambahkan objek user ke dalam list of users.
buat file dengan nama ***JavaReadXmlSaxEx.java***

```java
package Sesi8;

import java.util.List;

public class JavaReadXmlSaxEx {

    Run | Debug
    public static void main(String[] args) {

        MyRunner runner = new MyRunner();
        List<User> lines = runner.parseUsers();

        lines.forEach(System.out::println);
    }
}
```

Lalu coba build di terminal lalu masuk ke projectnya dan jalankan perintah :
mvn exec:java -Dexec.mainClass="com.ParsingXmlSax.module.Mail"

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.050 s
[INFO] Finished at: 2020-01-28T16:44:23+07:00
[INFO] ------------------------------------------------------------------------
[Maliks-Air:ParsingXmlSax swijaya$ mvn exec:java -Dexec.mainClass="com.ParsingXmlSax.module.Main"
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------< com.swijaya:ParsingXmlSax >-----------------------
[INFO] Building ParsingXmlSax 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO] 2
[INFO] --- exec-maven-plugin:1.6.0:java (default-cli) @ ParsingXmlSax ---
User{id=1, firstName=Peter, lastName=Brown, occupation=programmer}
User{id=2, firstName=Martin, lastName=Smith, occupation=accountant}
User{id=3, firstName=Lucy, lastName=Gordon, occupation=teacher}
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

Challenge

Challenge 1

Setelah anda mencoba teknik SAX sekarang coba selesaikan challenge berikut untuk jenis file xml dibawah ini :

```
<users>
    <user id="100">
        <firstname>Tom</firstname>
        <lastname>Hanks</lastname>
    </user>
    <user id="101">
        <firstname>Lokesh</firstname>
        <lastname>Gupta</lastname>
    </user>
    <user id="102">
        <firstnameHacktiv8firstname>
        <lastname>InJava</lastname>
    </user>
</users>
```

Ekspektasi Output :

```
Output:
[100:Tom:Hanks, 101:Lokesh:Gupta, 102:Hacktiv8:InJava]
```

Hint :
i. Create Model
ii. Build handler (DefaultParser)
iii. SAX Parser to read XML File
iv. Running your parser

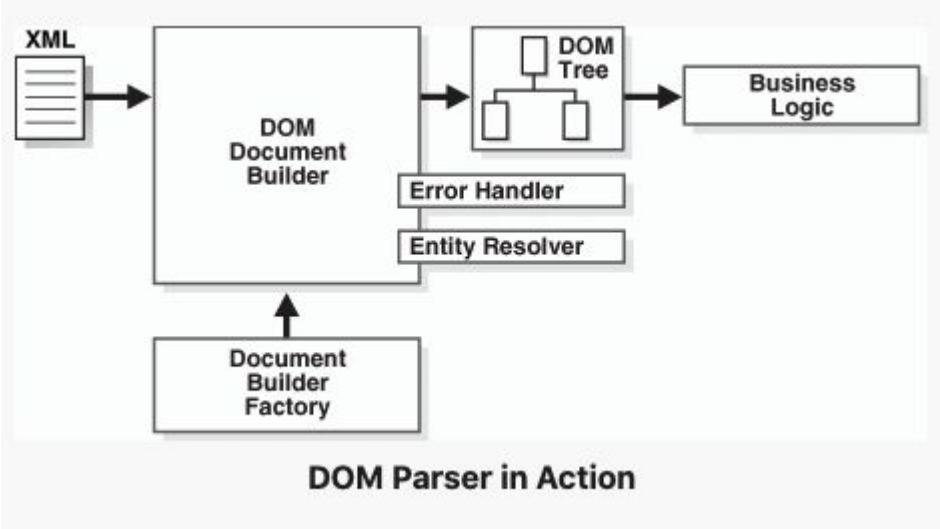
Parsing XML Data - DOM

Introduction

Nah jika kita sudah tau cara parse data dengan SAX, sekarang kita akan pelajari bagaimana parse data dengan DOM.

Apa itu DOM ?

DOM dikenal dengan Document Object Model, yang merupakan bagian dari Java API untuk XML Processing.



DOM Parser in Action

DOM Parser pada Java melintasi file XML dan membuat objek DOM yang sesuai.

Lalu Objek DOM ini dihubungkan kedalam struktur tree sehingga Parser membaca seluruh struktur XML ke dalam memori.

Untuk lebih memahami cara kerja kita lakukan praktik.

1. Buat contoh file xml lalu beri nama : employees.xml

```xml
<employees>
    <employee id="111">
        <firstName>Lokesh</firstName>
        <lastName>Gupta</lastName>
        <location>India</location>
    </employee>
    <employee id="222">
        <firstName>Alex</firstName>
        <lastName>Gussin</lastName>
        <location>Russia</location>
    </employee>
    <employee id="333">
        <firstName>David</firstName>
        <lastName>Feezor</lastName>
        <location>USA</location>
    </employee>
</employees>
```

Kedua kita buat Employee.java

```java
public class Employee
{
    private Integer id;
    private String firstName;
    private String lastName;
    private String location;

    @Override
    public String toString() {
        return "Employee [id=" + id + ", firstName=" + firstName
                + ", lastName=" + lastName + ", location=" + location + "]";
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getLocation() {
        return location;
    }

    public void setLocation(String location) {
        this.location = location;
    }
}
```

Penjelasan :

Step 1. Kita butuh import package DOM Parser pada aplikasi java kita.

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

Step 2. Buat DocumentBuilder

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

Step 3. Buat Dokumen Objek dari File XML

```
Document document = builder.parse(new File( file ));
```

Step 4. Validasi Struktur Dokumen

```
Schema schema = null;
try {
  String language = XMLConstants.W3C_XML_SCHEMA_NS_URI;
  SchemaFactory factory = SchemaFactory.newInstance(language);
  schema = factory.newSchema(new File(name));
  } catch (Exception e)
     {
     e.printStackStrace();
     }

Validator validator = schema.newValidator();
validator.validate(new DOMSource(document));
```

Step 5. Extract Root Element

```
Element root = document.getDocumentElement();
```

Step 6. Examines Attribute

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

Step 7. Examine Sub-Elements

```
node.getElementsByTagName("subElementName") //returns a list of sub-elements of specified name
node.getChildNodes()                        //returns a list of all child nodes
```

Step 8. Baca XML dengan DOM Parser

```
//Get Document Builder
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();

//Build Document
Document document = builder.parse(new File("employees.xml"));

//Normalize the XML Structure; It's just too important !!
document.getDocumentElement().normalize();

//Here comes the root node
Element root = document.getDocumentElement();
System.out.println(root.getNodeName());
```

```
 //Get all employees

NodeList nList = document.getElementsByTagName("employee");
System.out.println("============================");
for (int temp = 0; temp < nList.getLength(); temp++){
      Node node = nList.item(temp);
      System.out.println("");    //Just a separator

      if (node.getNodeType() == Node.ELEMENT_NODE)
      {
            //Print each employee's detail
            Element eElement = (Element) node;

System.out.println("Employee id : "    + eElement.getAttribute("id"));

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("Location : "    +
eElement.getElementsByTagName("location").item(0).getTextContent());

    }
}
```

Full Code: ***ParseKnownXMLStructure.java***

```java
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 ParseKnownXMLStructure {
    Run | Debug
    public static void main(String[] args) throws Exception {
        //Get Docuemnt Builder
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();

        //Build Document
        Document document = builder.parse(new File("employees.xml"));

        //Normalize the XML Structure; It's just too important !!
        document.getDocumentElement().normalize();

        //Here comes the root node
        Element root = document.getDocumentElement();
        System.out.println(root.getNodeName());

        //Get all employees
        NodeList nList = document.getElementsByTagName("employee");
        System.out.println("============================");

        for (int temp = 0; temp < nList.getLength(); temp++)
        {
         Node node = nList.item(temp);
         System.out.println("");    //Just a separator
         if (node.getNodeType() == Node.ELEMENT_NODE)
         {
            //Print each employee's detail
            Element eElement = (Element) node;
            System.out.println("Employee id : "    + eElement.getAttribute("id"));
            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("LOcation : "    + eElement.getElementsByTagName("location").item(0).getTextContent());
         }
        }
    }
}
```

HACKTIV8

Jalankan :

```
employees
============================

Employee id : 111
First Name : Lokesh
Last Name : Gupta
Location : India

Employee id : 222
First Name : Alex
Last Name : Gussin
Location : Russia

Employee id : 333
First Name : David
Last Name : Feezor
Location : USA
```

Case 2 : How to Parse Alexa API XML Response

Pada contoh sebelumnya kita selalu membuat file xml dari awal, nah pada case kali ini kita akan sedikit explorasi dengan apabila data xml bisa kita akses dari api public dalam hal ini alexa.

1. Buka url https://data.alexa.com/data?cli=10&url=https://bcafinance.co.id/ pada browser anda.


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--  Need more Alexa data?  Find our APIs here: https://aws.amazon.com/marketplace/seller-profile?id=4a9dbf38-88b1-4e87-a459-271154a77d2e  -->
▼<ALEXA VER="0.9" URL="bcafinance.co.id/" HOME="0" AID="=" IDN="bcafinance.co.id/">
 ▼<SD>
    <POPULARITY URL="bcafinance.co.id/" TEXT="865681" SOURCE="panel"/>
    <REACH RANK="813630"/>
    <RANK DELTA="+69565"/>
    <COUNTRY CODE="ID" NAME="Indonesia" RANK="11955"/>
  </SD>
 </ALEXA>
```

1. Lihat pada gambar diatas bahwa : Alexa API akan mengembalikan response XML ketika kita melakukan request pada url yang dimana didalam data XML terdapat XML Ranking termasuk juga element Popularity dari website bcafinance.co.id

3. Kita gunakan DOM Parser disini untuk langsung memilih element POPULARITY dan print out value dari atribut TEXT nya.

```
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("bcafinance.co.id");

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

    }
```

```java
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 bcafinance.co.id ranked 813630.

```
Terminal

Ranking: 813630
```