

XML Documents Processing Using SAX in Java

Lect. Mira SLAVU, lect.dr. Marian CRISTESCU
Universitatea „Lucian Blaga” Sibiu

XML, the eXtensible Markup Language has an enormous potential for the design and development of enterprise applications. As the W3C (World Wide Web Consortium) endorsed standard for document markup, XML defines a generic syntax to describe data with simple yet flexible tags included within documents. The packages required for XML documents processing are included in JDK beginning with the 1.4 version. SAX (Simple API for XML Parsing) allows the XML documents processing through some events. The processing modality is easy: the file is travel through, and at the XML elements appearance will be launch events which can be treat.

Keywords: documents XML, SAX, parser, SGML, CSS.

Introducere

XML (eXtensible Markup Language) este o metodă utilizată pentru schimbul datelor într-o rețea și este un subset al limbajului SGML (Standard Generalized Markup Language). Ca diferențe între HTML și XML amintesc faptul că XML a fost proiectat pentru descrierea datelor și este focalizat asupra structurii acestora, iar HTML a fost proiectat pentru afișarea datelor in browser.

Prezentarea unei oferte turistice într-un fișier în format XML

Fișierul *oferta_turistica.xml* conține următoarele elemente: cod_sejur, tara, statiune_localitate, hotel, clasificare, perioada, recomandari, de exemplu pentru Italia/Venetia: Piata San Marco, bazilica San Marco, Palatul Dogilor, canalul Grande, insula Burano.

```
<?xml version = "1.0" ?>
<sejururi>
<sejur>
  <cod_sejur> GR01 </cod_sejur>
  <tara> GRECIA </tara>
  <statiune_localitate> ATENA </statiune_localitate>
  <hotel> OLYMPIA </hotel>
  <clasificare> *** </clasificare>
  <perioada_inceput="01/08/2004" sfarsit="10/08/2004" />
  <recomandari>
    <obiective> PARTHENON </obiective>
    <obiective> PALATUL PRESEDINTELUI </obiective>
    <obiective> STADIONUL OLIMPIC </obiective>
  </recomandari>
</sejur>
<sejur>
  <cod_sejur> GR02 </cod_sejur>
  <tara> GRECIA </tara>
  <statiune_localitate> METEORA </statiune_localitate>
  <hotel> HERA </hotel>
  <clasificare> **** </clasificare>
  <perioadainceput="02/08/2004" sfarsit="08/08/2004" />
  <recomandari>
    <obiective> MANASTIREA METAMORFOSIS </obiective>
    <obiective> KALAMBAKA </obiective>
    <obiective> STANCILE METEOREI </obiective>
  </recomandari>
</sejur>
<sejur>
  <cod_sejur> TR01 </cod_sejur>
  <tara> TURCIA </tara>
  <statiune_localitate>EFES </statiune_localitate>
  <hotel> TAHSIN </hotel>
  <clasificare> *** </clasificare>
  <perioadainceput="01/08/2004" sfarsit="12/08/2004" />
  <recomandari>
    <obiective> TEMPLUL LUI HADRIAN </obiective>
    <obiective> POARTA LUI HERCULE </obiective>
    <obiective> AGORA </obiective>
    <obiective> FANTANA LUI TRAIAN </obiective>
  </recomandari>
```

```

</sejur>
<sejur>
  <cod_sejur> "IT01" </cod_sejur>
  < tara> ITALIA </ tara>
  < statiune_localitate> VENETIA </ statiune_localitate>
  < hotel> LAGUNA </ hotel>
  < clasificare> ***** </ clasificare>
  < perioada_inceput="05/08/2004" sfarsit="12/08/2004" />
  < recomandari>
    < obiective> PIATA SAN-MARCO </ obiective>
    < obiective> BAZILICA SAN MARCO </ obiective>
    < obiective> PALATUL DOGILOR </ obiective>
    < obiective> CANAL GRANDE </ obiective>
    < obiective> INSULA BURANO </ obiective>
  </ recomandari>
</sejur>
</sejururi>

```

Programul Java care procesează fișierul XML

Fișierul poate fi parsat destul de ușor pentru a extrage informațiile solicitate de un turist sau

care sunt necesare agenției. În programul următor sunt extrase principalele informații care privesc oferta turistică a unei țări solicitate de un turist.

```

// selecteaza din oferta cele care ofera sejururi intr-o anumit tar specificat
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.swing.*;
import javax.swing.tree.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;

public class AParser3 extends DefaultHandler{
  private final String strFileToParse = "c:\\oferta_turistica.xml";
  private final String strTaraSejur="ITALIA"; // tara pt. care se vor afisa
  ofertele
  String strTara = "";
  String strStatiuneLocalitate = "";
  String strHotel = "";
  String strClasificare = "";
  String strElement="";

  public AParser3(){
    SAXParser parserOferte=null;
    // metodele necesare pentru parsarea documentului sunt continute de clasa cu-
    renta
    DefaultHandler handlerFisier = this;
    try{
      // obtinerea parserului
      SAXParserFactory factory = SAXParserFactory.newInstance();
      parserOferte = (SAXParser)factory.newSAXParser();
    }catch(Exception e){
      e.printStackTrace();
      System.out.println("Eroare obtinere parser");
    }
    try {
      parserOferte.parse(strFileToParse,handlerFisier);
    }catch (Throwable t) {
      System.out.println("Eroare la parsare: Structura fisierului eronata");
      t.printStackTrace();
    }
  }
  public void startDocument()throws SAXException{
    System.out.println("***** START ANALIZA OFERTE *****");
  }
  public void endDocument() throws SAXException{
    System.out.println("***** SFARSIT ANALIZA OFERTE *****");
  }
  public void startElement(String namespace,
    String lName, // local name
    String qName, // qualified name
    Attributes attrs) throws SAXException
  {
    String eName = lName; // element name
    if (!"".equals(eName)) eName = qName; // namespaceAware = false
    strElement=eName;
  }
  public void endElement(String namespace,
    String lName, // local name
    String qName // qualified name
    throws SAXException {
  }
}

```

```

    public void characters(char buf[], int offset, int len) throws SAXException{
        String s = new String(buf, offset, len);
        if (strElement.equals("tara") && s.trim().length()>0)
            strTara=s; // memorez valoarea elementului "tara"
        if (strElement.equals("statiune_localitate")&& s.trim().length()>0)
            strStatiuneLocalitate=s; // memorez valoarea pt. localitate"
        if (strElement.equals("hotel") && s.trim().length()>0)
            strHotel=s; // memorez valoarea elementului "hotel"
        if (strElement.equals("clasificare") && s.trim().length()>0 ) {
            strClasificare=s; // memorez valoarea elementului "clasificare"
            if( strTara.trim().equals(strTaraSejur)){
                System.out.println("OFERTA:+strTaraSejur);
                System.out.println("LOCALITATE: +strStatiuneLocalitate);
                System.out.println(" HOTEL: "+strHotel);
                System.out.println("CLASIFICARE: "+strClasificare);
                System.out.println("");
            }
        }
    }
}
public static void main(String arg[]) {
    AParser3 aParser3 = new AParser3();
}
}

```

Concluzii

Limbajul Java oferă un bun suport pentru procesarea documentelor XML. Sunt disponibile ambele modalități de parsare – SAX și DOM. Fiecare dintre ele are avantaje și dezavantaje. SAX este utilizată pentru documente de mărime mare și care nu necesită acces aleatoriu la date. Parcurgerea este secvențială, de la început la sfârșit și nu presupune stocarea întregului document la un moment dat. DOM este frecvent utilizat în aplicațiile care necesită accesul la informații din diferite secțiuni ale documentului XML. Din cauza marelui consum de memorie, documentele XML nu trebuie să fie de mari dimensiuni.

Bibliografie

<http://java.sun.com/docs/books/tutorial>
www.w3.org/TR/1998/REC-xml-19980210
www.w3.org/TR/REC-xml-names
www.w3.org/TR/xPath
www.w3.org/TR/REC-html40
www.ietf.org/rfc