KXml XML Parser in Java ME

This article demonstrates how to use a XML parser in Java ME.

You can use XML parsers in Java ME applications to interface with an existing XML service. For example, you could get a customized view of news on your phone from an aggregator site that summarizes headlines and story descriptions for a news site in XML format.

XML parsers tend to be bulky, with heavy run-time memory requirements. In order to adapt to the MIDP environment, XML parsers must be small to meet the resource constraints of MIDP-based devices. They should also be easily portable, with minimum effort required to port them to MIDP.

Two frequently used XML parsers for resource-constrained devices are kXML and NanoXML.

kXML is a small XML pull parser, specially designed for constrained environments like Java ME platform (CLDC and MIDP).The current version of kXML is KXml2. In contrast to kXML1, it features cursor API instead of event objects, leading to a reduced footprint and less object creation overhead.

As of version 1.6.8 for MIDP, NanoXML supports DOM parsing.

There is also XML parser in the Web Services API (JSR-172), which can be used instead of the above mentioned parsers, which needs to be included in every MIDlet suite trying to use them. The XML parsing package has been implemented on S60 3rd Edition devices (and newer) as well as on Series 40 3rd Edition Feature Pack 1 devices (and newer).

Using KXml

Download the latest version of KXml2 here. Include the jar to your project lib.

Let us take a simple XML to be parsed.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <qset>
 <qn>
  <txt>Capital of India is</txt>
  <A>New Delhi</A>
  <B>Jaipur</B>
  <C>Mumbai</C>
  <D>Bengaluru</D>
  <ans>1</ans>
 </qn>
 <qn>
        <txt>India got independence in</txt>
  <A>1878</A>
  <B>1947</B>
  <C>1957</C>
  <D>2007</D>
  <ans>2</ans>
 </qn>
   </qset>
```

```
import java.io.IOException;
import java.util.Vector;
import org.kxml2.io.KXmlParser;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;

/**
 * Parses the XML file
 * @author shivam
```


(C) Copyright Nokia 2014. All rights reserved.

```
 */

public class Parser {

    /**
     * Kxml parser instance
     */
    private KXmlParser xmlParser = null;
    /**
     * singleton instance of Parser class
     */
    private static Parser instance = null;

    //private Vector questionSet = null;
    /** Creates a new instance of Parser */


    private Parser() {
        xmlParser = new KXmlParser();
        questionSet = new Vector(Utility.TOTAL_QUESTIONS);
    }

    /**
     * get the instance of this class
     * @return instance
     */
    public static Parser getInstance()
    {
        if(instance == null)
        {
            instance = new Parser();
        }
        return instance;

    }

    /**
     * Parses the XML file
     * @throws java.io.IOException error parsing
     */
    public void getQuestions() throws IOException
    {
        try
        {
            xmlParser.setInput(new ReadFile().gtFileStream(Utility.QUESTION_FILE_PATH));
            xmlParser.nextTag();
            xmlParser.require(XmlPullParser.START_TAG, null,Utility.TAG_QUESTION_SET);
            while (xmlParser.nextTag () != XmlPullParser.END_TAG)
            {
                questionSet.addElement(readXMLData(xmlParser));
            }
            xmlParser.require(XmlPullParser.END_TAG, null, Utility.TAG_QUESTION_SET);
            xmlParser.next();
            xmlParser.require(XmlPullParser.END_DOCUMENT, null, null);
        }

        catch(Exception ioe)
```


(C) Copyright Nokia 2014. All rights reserved.

```
        {
            System.err.println(ioe.getMessage());
            throw new IOException(ioe.getMessage());
        }

    }

    /**
     * Reads the XML data in a buffer
     * @param parser kxml parser object
     * @return Question
     * @throws java.io.IOException error reading data
     * @throws org.xmlpull.v1.XmlPullParserException parser exception
     */
     private Question readXMLData(KXmlParser parser) throws IOException,
XmlPullParserException {


   //Parse our XML file

   parser.require(XmlPullParser.START_TAG, null, Utility.TAG_QUESTION);
                       Question quest = new Question();

   while (parser.nextTag() != XmlPullParser.END_TAG) {

    parser.require(XmlPullParser.START_TAG, null, null);
    String name = parser.getName();

    String text = parser.nextText();

    System.out.println ("<"+name+">"+text);

    if (name.equals(Utility.TAG_QUESTION_DESCRIPTION))
     quest.setQuestionText(text);
                                else if (name.equals(Utility.TAG_QUESTION_ANSWER))
                                 quest.setAnswer((byte)(Integer.parseInt(text)));
                               else if (name.equals(Utility.TAG_QUESTION_OPTION_A))
     quest.setOptionA(text);
                                else if (name.equals(Utility.TAG_QUESTION_OPTION_B))
     quest.setOptionB(text);
                                else if (name.equals(Utility.TAG_QUESTION_OPTION_C))
     quest.setOptionC(text);
                                else if (name.equals(Utility.TAG_QUESTION_OPTION_D))
     quest.setOptionD(text);
                           parser.require(XmlPullParser.END_TAG, null, name);
   }

   parser.require(XmlPullParser.END_TAG, null, Utility.TAG_QUESTION);
                       return quest;
        }
  }
```

Helpful articles:

- http://developers.sun.com/mobility/midp/articles/parsingxml/

(C) Copyright Nokia 2014. All rights reserved.

- http://www.devx.com/xml/Article/11773
- http://www.ibm.com/developerworks/library/wi-parsexml/

(C) Copyright Nokia 2014. All rights reserved.