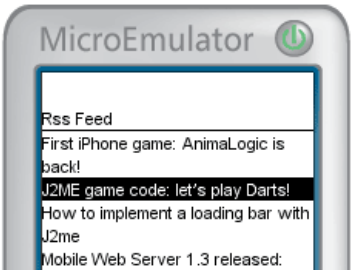
RSS Parser with KXML in Java ME

We'll build a simple RSS parser using the KXML library. You can download the full source code (including a sample test midlet). To see the test midlet in action you can go to here.



RssItem Class

This is the class that will represent a single Rss Item instance. Within this article, we'll only care about the title, link and description fields of RSS items, but including other fields will be easy once the process is clear.

```
public class RssItem
{
 public String title = null;
 public String description = null;
 public String link = null;

 public RssItem()
 {
 }

 public RssItem(String title, String link, String description)
 {
  this.title = title;
  this.link = link;
  this.description = description;
 }
}
```

KXmlRssParser Class

This is the class that will actually parse the RSS feed.

The only public method will be parse(String rssUrl), accepting the RSS feed's URL as the argument, and returning a Vector containing the parsed RssItems.

```
public ArrayList parse(String rssUrl) throws Exception
{
 ArrayList items = new ArrayList();

 KXmlParser parser = new KXmlParser();

 HttpConnection conn = (HttpConnection)Connector.open(rssUrl);

 InputStream rssStream = conn.openInputStream();

 InputStreamReader isr = new InputStreamReader(rssStream);
```


(C) Copyright Nokia 2014. All rights reserved.

```
 parser.setInput(isr);

 parser.nextTag();

 parser.require(XmlPullParser.START_TAG, null, "rss");

 parser.nextTag();

 parser.require(XmlPullParser.START_TAG, null, "channel");

 parser.nextTag();

 while(parser.getEventType() != XmlPullParser.END_TAG)
 {
  String nodeName = parser.getName();

  if(nodeName.equals("item"))
  {
   items.add(parseRssItem(parser));
  }
  else
  {
   parser.skipSubTree();
  }
  parser.nextTag();
 }
 isr.close();

 rssStream.close();

 conn.close();

 return items;
}
```

We have first instantiated our items Vector and KXmlParser. Then we have set the parser input with an InputStream, via an InputStreamReader. Then, we call 2 times the nextTag() method, and check for "rss" and "channel" tags to be present:

```
parser.nextTag();

parser.require(XmlPullParser.START_TAG, null, "rss");

parser.nextTag();

parser.require(XmlPullParser.START_TAG, null, "channel");
```

Then, after calling once again nextTag(), we enter a while loop and check for all available "item" tags, discarding all other information with the skipSubTree() method.

```
while(parser.getEventType() != XmlPullParser.END_TAG)
{
 String nodeName = parser.getName();

 if(nodeName.equals("item"))
```


(C) Copyright Nokia 2014. All rights reserved.

```
 {
  items.add(parseRssItem(parser));
 }
 else
 {
  parser.skipSubTree();
 }
 parser.nextTag();
}
```

When we encounter a "item" tag, we parse it with the following parseRssItem() method. This method, as seen before, will parse only title, link and description information, discarding all other data.

```
RssItem parseRssItem(KXmlParser parser) throws Exception
{
 RssItem item = new RssItem();

 parser.nextTag();

 while(parser.getEventType() != XmlPullParser.END_TAG)
 {
  String nodeName = parser.getName();

  if(nodeName.equals("title"))
  {
   item.title = parser.nextText();
  }
  else if(nodeName.equals("description"))
  {
   item.description = parser.nextText();
  }
  else if(nodeName.equals("link"))
  {
   item.link = parser.nextText();
  }
  else
  {
   parser.skipSubTree();
  }
  parser.nextTag();
 }
 return item;
}
```

Finally, we have to close all our open resources: the InputStreamReader, InputStream and Connection instances.

Simple Usage of KXmlRssParser class

We can build simple list of rss item titles with the following code:

```
public TitleList()
{
 super("Rss Feed", List.IMPLICIT);

 KXmlRssParser parser = new KXmlRssParser();
```


(C) Copyright Nokia 2014. All rights reserved.

```
 try
 {
  ArrayList rssItems = parser.parse("http://www.jappit.com/blog/feed/");

  for(Iterator it = rssItems.iterator(); it.hasNext(); )
  {
   append(((RssItem)it.next()).title, null);
  }
 }
 catch(Exception e)
 {
  append("Error: " + e, null);

  e.printStackTrace();
 }
}
```


(C) Copyright Nokia 2014. All rights reserved.