BSF4ooRexx

Parsing XML Documents with SAX (Simple API for XML)

Business Programming 2

Windows GUIs (AWT)	Sockets SSL/TLS	XML SAX/DOM JSON	Scripting AOO/LO (UNO)	Rexx Script Engine	Portable GUIs (JavaFX)	Java Web Server (Tomcat)	Java Classes written in Rexx style

Prof. Rony G. Flatscher

Markup Language

- Text, marked up in HTML

```
<html>
  <head>
     <title>This is my HTML file</title>
  </head>
  <body>
     <h1>Important Heading</h1>
     <p>This <span class="verb">is</span> the
        first paragraph.</p>
     <h1>Another Important Heading</h1>
     <p id="xyz1">Another paragraph.</p>
     <p id="a9876">This <span class="verb">is</span> it.</p>
  </body>
</html>
```

Web Browser Output:

Important Heading

This is the first paragraph.

Another Important Heading

Another paragraph.

This is it.

SAX (Simple API for XML), 1

- A SAX parser sequentially parses a XML document
- The Java SAX parser interfaces are defined in the package org.xml.sax
- Each time a meaningful piece of characters got parsed, the SAX parser will inform registered listener objects
  - The SAX parser available with Java defines the methods listener objects must implement: org.xml.sax.ContentHandler
    - Search on the Internet either with the string "*javadoc 8 org.xml.sax.ContentHandler*" or "*javadoc ContentHandler*"
  - Each method represents one "SAX event", including the argument a SAX parser supplies to listener objects

SAX (Simple API for XML), 2

- A SAX parser informs registered SAX event listener objects about the following SAX parsing events (in the following order)
  - setDocumentLocator(Locator locator)
  - ➔ **startDocument()**
  - startPrefixMapping(String prefix, String uri)
    - ➔ skippedEntity(String name)
    - ➔ **startElement(**String uri, String localName, String qName, Attributes atts**)**
    - ➔ ignorableWhitespace(char[] ch, int start, int length)
    - ➔ **characters(**char[] ch, int start, int length**)**
    - ➔ **endElement(**String uri, String localName, String qName**)**
  - endPrefixMapping(String prefix)
  - ➔ **endDocument()**

SAX (Simple API for XML), 3

- The interface org.xml.sax.ErrorHandler defines the methods a SAX/DOM error listener must implement
  - error(SAXParseException exception)
  - fatalError(SAXParseException exception)
  - warning(SAXParseException exception)
- org.xml.sax.SAXParseException has the following methods
  - getCause() returns a Throwable Java object representing the cause
  - getException() returns an embedded exception, if any
  - getMessage() returns a string with the detailed error message
  - toString() returns a string representation of the SAXParseException

Defining a SAX Listener in ooRexx

- Create an ooRexx listener class
  - For each SAX event you wish to process, create an ooRexx method by the same name and fetch the arguments, if any, using USE ARG
  - If SAX events are intentionally not handled, then define a method named UNKNOWN, such that Rexx does not raise a condition
- Create an ooRexx listener object from it
- Create a Java object that embeds the ooRexx listener object
  - BSFCreateRexxProxy(rexxListenerObject,[slotArg],interfaceName[,...])
  - interfaceName denotes the Java interface name which methods the Rexx listener object handles
    - It is possible to denote more than one Java interface, if the Rexx listener object is able to handle all methods defined by them!

Extract Text From Any XHTML Document (1/2)

```
parse arg xmlFileName
rexxObject=.saxHandler~new -- create a Rexx SAX handler object
   -- wrap up the Rexx SAX handler as a Java object
javaProxy=BSFCreateRexxProxy(rexxObject,,"org.xml.sax.ContentHandler")

   -- create a Java SAX parser object and register our content handler object
parser=bsf.loadClass("org.xml.sax.helpers.XMLReaderFactory")~createXMLReader
parser~setContentHandler(javaProxy) -- set the content handler for this parser

eh=.errorHandler~new       -- create an error handler Rexx object
   -- wrap up the Rexx error handler as a Java object
javaEH=BsfCreateRexxProxy(eh, , "org.xml.sax.ErrorHandler")
parser~setErrorHandler(javaEH)   -- set the error handler for this parser

parser~parse(xmlFileName)  -- parse the InputStream, will call back

::requires BSF.CLS         -- get the Java support for ooRexx

::class "SaxHandler"    -- a Rexx content handler ("org.xml.sax.ContentHandler")
::method characters        -- the callback method for characters (text)
  use arg textCharArray, start, length -- arguments from the Java SAX parser
  say pp(.bsf~new("java.lang.String", textCharArray, start, length)~toString)
::method unknown           -- intercept all other messages to avoid runtime error

::class ErrorHandler    -- a Rexx error handler ("org.xml.sax.ErrorHandler")
::method unknown        /* handles "warning", "error" and "fatalError" events */
  use arg methName, argArray  -- arguments from the Java SAX parser
  exception=argArray[1] /* retrieve SAXException argument               */
  .error~say(methName":" "line="exception~getLineNumber",col="exception~getColumnNumber":" pp(exception~getMessage))
```

Extract Text From Any XHTML Document (2/2)

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html>
  <head>
     <title>This is my HTML file</title>
     <link rel="stylesheet" type="text/css" href="example2.css"/>
  </head>
  <body>
     <h1>Important Heading</h1>
     <p>This <span class="verb">is</span> the
        first paragraph.</p>
     <h1>Another Important Heading</h1>
     <p id="xyz1">Another paragraph.</p>
     <p id="a9876">This <span class="verb">is</span> it.</p>
  </body>
</html>
```

```
$ rexx sax_01.rxj example2.html
```

Output:

```
[This is my HTML file]
[
     ]
[Important Heading]
[
     ]
[This ]
[is]
[ the
        first paragraph.]
[
     ]
[Another Important Heading]
[
     ]
[Another paragraph.]
[
     ]
[This ]
[is]
[ it.]
[
  ]
```

List Elements in Document Order (1/2)

```
parse arg xmlFileName
rexxObject=.saxHandler~new -- create a Rexx SAX handler object
   -- wrap up the Rexx SAX handler as a Java object
javaProxy=BSFCreateRexxProxy(rexxObject,,"org.xml.sax.ContentHandler")

   -- create a Java SAX parser object and register our content handler object
parser=bsf.loadClass("org.xml.sax.helpers.XMLReaderFactory")~createXMLReader
parser~setContentHandler(javaProxy) -- set the content handler for this parser

eh=.errorHandler~new       -- create an error handler Rexx object
   -- wrap up the Rexx error handler as a Java object
javaEH=BsfCreateRexxProxy(eh, , "org.xml.sax.ErrorHandler")
parser~setErrorHandler(javaEH)   -- set the error handler for this parser

parser~parse(xmlFileName)  -- parse the InputStream, will call back

::requires BSF.CLS         -- get the Java support for ooRexx

::class "SaxHandler"    -- a Rexx content handler ("org.xml.sax.ContentHandler")
::method startElement      -- the callback method for characters (text)
  use arg  , localName
  say pp(localName)
::method unknown           -- intercept all other messages to avoid runtime error

::class ErrorHandler    -- a Rexx error handler ("org.xml.sax.ErrorHandler")
::method unknown        /* handles "warning", "error" and "fatalError" events */
  use arg methName, argArray  -- arguments from the Java SAX parser
  exception=argArray[1] /* retrieve SAXException argument               */
  .error~say(methName":" "line="exception~getLineNumber",col="exception~getColumnNumber":" pp(exception~getMessage))
```

Changes only in the class!

List Elements in Document Order (2/2)

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html>
  <head>
     <title>This is my HTML file</title>
     <link rel="stylesheet" type="text/css" href="example2.css"/>
  </head>
  <body>
     <h1>Important Heading</h1>
     <p>This <span class="verb">is</span> the
        first paragraph.</p>
     <h1>Another Important Heading</h1>
     <p id="xyz1">Another paragraph.</p>
     <p id="a9876">This <span class="verb">is</span> it.</p>
  </body>
</html>
```

```
$ rexx sax_02.rxj example2.html
```

Output:

```
[html]
[head]
[title]
[link]
[body]
[h1]
[p]
[span]
[h1]
[p]
[p]
[span]
```

List Elements in Document Order, Indented (1/2)

```
parse arg xmlFileName
rexxObject=.saxHandler~new -- create a Rexx SAX handler object
   -- wrap up the Rexx SAX handler as a Java object
javaProxy=BSFCreateRexxProxy(rexxObject,,"org.xml.sax.ContentHandler")

   -- create a Java SAX parser object and register our content handler object
parser=bsf.loadClass("org.xml.sax.helpers.XMLReaderFactory")~createXMLReader
parser~setContentHandler(javaProxy) -- set the content handler for this parser

eh=.errorHandler~new       -- create an error handler Rexx object
   -- wrap up the Rexx error handler as a Java object
javaEH=BsfCreateRexxProxy(eh, , "org.xml.sax.ErrorHandler")
parser~setErrorHandler(javaEH)   -- set the error handler for this parser

parser~parse(xmlFileName)  -- parse the InputStream, will call back

::requires BSF.CLS         -- get the Java support for ooRexx

::class "SaxHandler"    -- a Rexx content handler ("org.xml.sax.ContentHandler")
::method init              -- ooRexx constructor
  expose level             -- object attribute (variable)
  level=0                  -- initialize to 0
::method startElement      -- the callback method for characters (text)
  expose level
  use arg  , localName
  say "   "~copies(level) || pp(localName)
  level+=1                 -- increase level by 1
::method endElement
  expose level
  level-=1                 -- decrease level by 1
::method unknown           -- intercept all other messages to avoid runtime error

::class ErrorHandler    -- a Rexx error handler ("org.xml.sax.ErrorHandler")
… cut …
```

Changes only in the class!

List Elements in Document Order, Indented (2/2)

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html>
  <head>
     <title>This is my HTML file</title>
     <link rel="stylesheet" type="text/css" href="example2.css"/>
  </head>
  <body>
     <h1>Important Heading</h1>
     <p>This <span class="verb">is</span> the
        first paragraph.</p>
     <h1>Another Important Heading</h1>
     <p id="xyz1">Another paragraph.</p>
     <p id="a9876">This <span class="verb">is</span> it.</p>
  </body>
</html>
```

```
$ rexx sax_03.rxj example2.html
```

Output:

```
[html]
   [head]
      [title]
      [link]
   [body]
      [h1]
      [p]
         [span]
      [h1]
      [p]
      [p]
         [span]
```

List Elements with Text (1/1)

```
parse arg xmlFileName
rexxObject=.saxHandler~new -- create a Rexx SAX handler object
   -- wrap up the Rexx SAX handler as a Java object
javaProxy=BSFCreateRexxProxy(rexxObject,,"org.xml.sax.ContentHandler")

   -- create a Java SAX parser object and register our content handler object
parser=bsf.loadClass("org.xml.sax.helpers.XMLReaderFactory")~createXMLReader
parser~setContentHandler(javaProxy) -- set the content handler for this parser

eh=.errorHandler~new       -- create an error handler Rexx object
   -- wrap up the Rexx error handler as a Java object
javaEH=BsfCreateRexxProxy(eh, , "org.xml.sax.ErrorHandler")
parser~setErrorHandler(javaEH)   -- set the error handler for this parser

parser~parse(xmlFileName)  -- parse the InputStream, will call back

::requires BSF.CLS         -- get the Java support for ooRexx

::class "SaxHandler"    -- a Rexx content handler ("org.xml.sax.ContentHandler")
::method init              -- ooRexx constructor
  expose level             -- establish direct access to attribute
  level=0                  -- initialize to 0
::method startElement      -- the callback method for characters (text)
  expose level             -- establish direct access to attribute
  use arg  , localName
  say "   "~copies(level) || pp(localName)
  level+=1                 -- increase level by 1
::method endElement
  expose level             -- establish direct access to attribute
  level-=1                 -- decrease level by 1
::method characters        -- the callback method for characters (text)
  expose level             -- establish direct access to attribute
  use arg textCharArray, start, length -- arguments from the Java SAX parser
  say "   "~copies(level) "-->" pp(.bsf~new("java.lang.String", textCharArray, start, length)~toString)
::method unknown           -- intercept all other messages to avoid runtime error

::class ErrorHandler    -- a Rexx error handler ("org.xml.sax.ErrorHandler")
...cut...
```

Changes only in the class!

List Elements with Text (2/2)

HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html>
  <head>
     <title>This is my HTML file</title>
     <link rel="stylesheet" type="text/css" href="example2.css"/>
  </head>
  <body>
     <h1>Important Heading</h1>
     <p>This <span class="verb">is</span> the
        first paragraph.</p>
     <h1>Another Important Heading</h1>
     <p id="xyz1">Another paragraph.</p>
     <p id="a9876">This <span class="verb">is</span> it.</p>
  </body>
</html>
```

```
$ rexx sax_04.rxj example2.html
```

Output:

```
[html]
   [head]
      [title]
          --> [This is my HTML file]
      [link]
   [body]
       --> [
     ]
      [h1]
          --> [Important Heading]
       --> [
     ]
      [p]
          --> [This ]
         [span]
             --> [is]
          --> [ the
       first paragraph.]
       --> [
     ]
      [h1]
          --> [Another Important Heading]
       --> [
     ]
      [p]
          --> [Another paragraph.]
       --> [
     ]
      [p]
          --> [This ]
         [span]
             --> [is]
          --> [ it.]
       --> [
  ]
```

Roundup

- Parsing any XML encoded document possible
  - Using BSF4ooRexx
  - Exploiting Java's functionality for parsing XML documents
- SAX parsing
  - SAX parser defines events
  - SAX parser invokes the respective SAX event method in the registered callback object
  - Concepts quite easy, memory efficient
- Easy to exploit from ooRexx !

Further Information

- World Wide Web Consortium ("W3C")
  - <https://www.w3.org/> (2022-12-12)
    - <https://www.w3.org/Style/CSS/>
    - <https://dom.spec.whatwg.org/>
    - <https://www.w3.org/MarkUp/>
    - <https://www.w3.org/QA/2002/04/valid-dtd-list.html>
- SAX specific URLs (2022-12-12)
  - <http://www.saxproject.org/>
  - <http://www.cafeconleche.org/books/xmljava/chapters/index.html>
  - <https://docs.oracle.com/javase/7/docs/api/org/xml/sax/package-summary.html>