Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-nine)

## 9.6: Parsing with StAX

StAX is an event based, streaming API. It reads an XML file part by part, instead of the entire file at once as with DOM. After it reads a part of an XML document, it sends an event code. Event codes include DTD, START_ELEMENT, CHARACTERS and END_ELEMENT (see the documentation on the XMLStreamConstants interface for a complete list). It is up to the programmer to implement the handling each event code.

As you can see, StAX offers flexibility but requires more coding. On the other hand, it can be much faster, especially if you need to read only parts of large files. It also gives you flexibility on how you store the data (or not to store it at all if that is what your application calls for).

The example below shows how to parse the dvd.xml file and display its content on the screen:

```
import javax.xml.stream.*;
import java.io.*;
import java.util.*;

public class StaxDVDReader {
 public void read(String  filepath) {
 try {
  FileInputStream fileInputStream = new  FileInputStream(filepath);
       XMLInputFactory factory = XMLInputFactory.newInstance();
       XMLStreamReader xmlStreamReader =
           factory.createXMLStreamReader(fileInputStream);
  while (xmlStreamReader.hasNext()) {
   readDVDList(xmlStreamReader);
  }
  xmlStreamReader.close();
  } catch  (XMLStreamException e) {
       e.printStackTrace();
  } catch  (FileNotFoundException e) {
       e.printStackTrace();
  }
 }
 private void  readDVDList(XMLStreamReader reader) throws XMLStreamException {
 int eventCode =  reader.next();
 switch (eventCode) {
  case XMLStreamReader.START_ELEMENT:
       String key = reader.getLocalName();
       if (key.equals("DVD")) {
            readDVD(reader);
       }
   break;
  }
 }

 private void  readDVD(XMLStreamReader reader)
       throws XMLStreamException {
 String name = "";
 String value = "";

 int nAttributes =  reader.getAttributeCount();
 String avalue =  reader.getAttributeValue(0);
 System.out.println("attribute  value: " + avalue + " attribute count: " + nAttributes);

 while (reader.hasNext()) {
  int eventCode = reader.next();
       switch (eventCode) {
       case XMLStreamReader.START_ELEMENT:
            name =  reader.getLocalName();
            break;
       case XMLStreamReader.END_ELEMENT:
            name =  reader.getLocalName();
            if (name.equals("DVD"))  return;
            break;
       case XMLStreamReader.CHARACTERS:
        value =  reader.getText();
            System.out.println("Element  name= " + name + " value=" + value);
            break;
       }
 }
 return;
 }

 public static void  main(String[] args) {
 StaxDVDReader dvdReader =  new StaxDVDReader();
 dvdReader.read("dvd.xml");
 }
}
```

Here are steps in writing code to parse an XML document with StAX.

1. Import the following libraries:

```
    import javax.xml.stream.*;
    import java.io.*;
```

2. Create an XMLInputFactory . See the read( ) method above.
3. Create an XMLStreamReader and pass a Reader to it such as a FileReader. The XML file is passed as a parameter to FileReader.
4. We can now iterate through the contents of our XML file using the streamreader's next( ) method.
5. next( ) returns an event code that indicates which part of the document has been read such as: DTD, START_ELEMENT, CHARACTERS and END_ELEMENT.
6. If you get the START_ELEMENT event code, you can retrieve the element's name using the getLocalName( ) method. To read the attributes, use getAttributeValue( ) method.
7. To read the text between the start and end tags, wait until you receive the CHARACTERS event code. Afterwards, you can read the text using getText( ).

In our example, the readDVDList( ) method looks for elements that start with the DVD tag and passes the stream to the readDVD( ) method. readDVD( ) reads everything under the DVD element. If you want to copy the data to a custom data structure, this is the place to do it.