Unit 1-part II: access XML file with SAX

## OBJECTIVE:

OPEN AN XML FILE USING SAX AND TRAVERSE IT, SHOWING ALL THE ELEMENTS

## PRACTICE:

1. **Extend class DefaultHandler: create class mySAXhandler.**
2. **Override the methods startElement,endElement and characters**

**import** org.xml.sax.Attributes;

**import** org.xml.sax.helpers.DefaultHandler;

**import** org.xml.sax.\*;

**public** **class** mySAXhandler **extends** DefaultHandler{

**int** lastElement;//used to know what type of tag is generating the event

String result="";//stores the result. We will use it later

**public** mySAXhandler() {

lastElement=0;//

}

@Override

**public** **void** startElement(String uri,String localName,String qName,Attributes atts) **throws** SAXException{

//code here

}

@Override

**public** **void** endElement(String uri, String localName,String qName) **throws** SAXException{

//code here

}

@Override

**public** **void** characters(**char**[] ch, **int** start, **int** length) **throws** SAXException{

//code here

}

}

**We write our code:**

**import** org.xml.sax.Attributes;

**import** org.xml.sax.helpers.DefaultHandler;

**import** org.xml.sax.\*;

**public** **class** mySAXhandler **extends** DefaultHandler{

**int** lastElement;//used to know what type of tag is generating the event

**public** mySAXhandler() {

lastElement=0;//

}

@Override

**public** **void** startElement(String uri,String localName,String qName,Attributes atts) **throws** SAXException{

**if**(qName.equals("Book")){

System.***out***.println("Published in: "+atts.getValue(atts.getQName(0)));//extracts the first attribute

lastElement=1;

}**else** **if**(qName.equals("Title")){

System.***out***.print("\n "+"The title es: ");//we will know the value of Title later, in event characters

lastElement=2;

}**else** **if**(qName.equals("Author")){

System.***out***.print("\n "+"The author is: ");

lastElement=3;

}

}

@Override

**public** **void** endElement(String uri, String localName,String qName) **throws** SAXException{

**if**(qName.equals("Book")){

System.***out***.println("\n-----------------------");

}

}

@Override

**public** **void** characters(**char**[] ch, **int** start, **int** length) **throws** SAXException{

String car=**new** String(ch,start,length);

car=car.replaceAll("\t","");//cleans all tabs System.***out***.print(car); }

}

1. **Use your SAX Handler in the class with your SAX methods ( create class AccesXMLSAX)**

**import** java.io.File;

**import** javax.xml.parsers.SAXParser;

**import** javax.xml.parsers.SAXParserFactory;

**public** **class** AccessXMLSAX {

mySAXhandler sh;

SAXParser parser;

**public** **int** openXMLwithSAX(){

**try**{

SAXParserFactory factory=SAXParserFactory.*newInstance*();

parser=factory.newSAXParser();

sh=**new** mySAXhandler();

**return** 0;

}**catch**(Exception e){

e.printStackTrace();

**return** -1;

}

}

**public** **void** traverseSAX(File fXML){

**try**{

parser.parse(fXML, sh);

}**catch**(Exception e){

e.printStackTrace();

System.***out***.println("Error al parsear con SAX");

}

}

}

1. **Create a new class to use your class AccessXMLSAX**

**import** javax.xml.parsers.SAXParser;

**import** java.io.\*;

**public** **class** UseAccessXMLSAX {

**public** **static** **void** main(String[] args) {

File fXML=**new** File("Books.xml");

AccessXMLSAX a =**new** AccessXMLSAX();

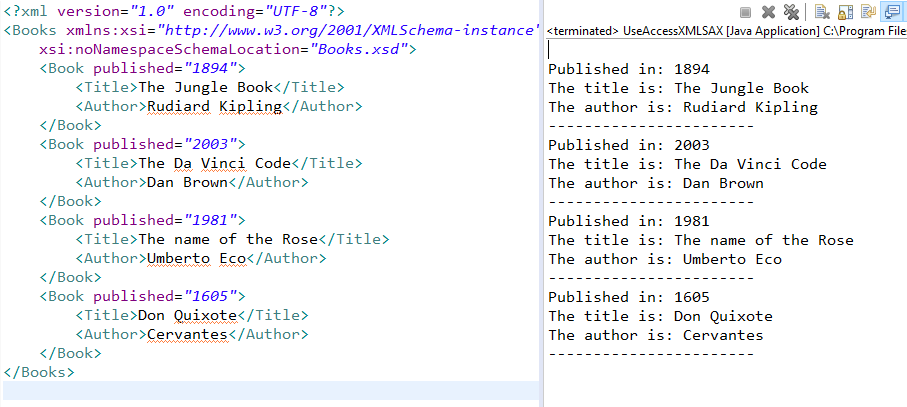
a.openXMLwithSAX();

a.traverseSAX(fXML);

}

}

1. When you run it, you must get something like this:



1. **We can use the class XMLReader to obtain the same thing:**

**public** **void** OpenTraverseSAXUsingXMLReader(String fileName) **throws** SAXException, IOException {

System.***out***.println("\*\*\*\*\*\* Solution using XMLReader \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

XMLReader xmlr=XMLReaderFactory.*createXMLReader*();

sh=**new** mySAXhandler();

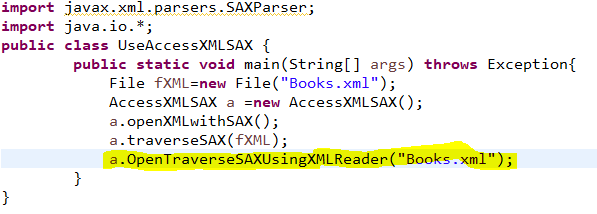
xmlr.setContentHandler(sh);

InputSource fileXML= **new** InputSource(fileName);

xmlr.parse(fileXML);//parse generating output with the events

}

**Then we can use than method:**



**And the output is:**

