

Internet Programming with XML

Java and SAX

Module Objectives

- Create a SAX parser in Java
- Use the event-driven methods of DefaultHandler
- Handle the component nodes in SAX
- Parse & display an entire document
- Filter XML documents
- Create a windowed browser
- Navigate & modify XML Documents

Reading Assignments:

- **Textbook:** chapters 13, 15, 16
-

The Simple API for XML (SAX) APIs

Interfaces	Uses
Simple API for XML Processing (SAX)	• XML to XML • non-XML to XML • XML to non-XML

The "[Simple API](#)" for XML (SAX) is the event-driven, serial-access mechanism that does element-by-element processing. The API for this level reads and writes XML to a data repository or the Web.

Event-based

- [SAX Project](#)
 - SAX
 - SAX2

Produce SAX events. Parse existing XML document, or produce SAX events from non-XML data.

Consume SAX events. Process or filter XML data.

- [SAX tutorial](#)
- [SAX sample](#)

The DefaultHandler Class

A [DefaultHandler](#), defined in the `org.xml.sax.helpers` package,

implements the [ContentHandler](#), [ErrorHandler](#), [DTDHandler](#), and [EntityResolver](#) interfaces (with null methods), so you can override only the ones you're interested in.

Implementing the ContentHandler Interface

The [ContentHandler](#) interface requires a number of methods that the SAX parser invokes in response to different parsing events. Methods like `startDocument`, `endDocument`, `startElement`, and `endElement` are invoked when an XML tag is recognized. This interface also defines methods `characters` and `processingInstruction`, which are invoked when the parser encounters the text in an XML element or an inline processing instruction, respectively.

The easiest way to implement this interface is to extend the [DefaultHandler](#) class.

SAX Events

- `startDocument`
- `endDocument`
- `startElement`
- `endElement`
- `characters`
- `ignorableWhitespace`
- `startPrefixMappings`
- `endPrefixMappings`
- `processingInstruction`

Creating a Parser and Parsing a Document

```
// Use an instance of this class as the SAX event handler
DefaultHandler handler = this;

// Use the default (non-validating) parser
SAXParserFactory factory = SAXParserFactory.newInstance();
try
{
    // Parse the input file using this DefaultHandler object
    SAXParser saxParser = factory.newSAXParser();
    saxParser.parse( new File(url), handler );
}
catch (Throwable t)
{
    t.printStackTrace();
}
```

Resources:

- [JAXP Reference](#)
- [Java API for XML Processing \(JAXP\)](#)
- [Java API for XML Processing Tutorial](#)
- [SAX Tutorial](#)
- [Java 2 Platform, Standard Edition, v 1.4.2 API Specification](#)
- [javax.xml.parsers](#)
- [org.xml.sax](#)

- org.xml.sax.ext
- org.xml.sax.helpers
- javax.xml.transform.sax

Hands-on Assignment:

- [Assignment 4](#) due **Week 8**