StAX

StAX API

Introduction

The primary goal of the StAX API is to give "parsing control to the programmer by exposing a simple iterator based API. This allows the programmer to ask for the next event (pull the event) and allows state to be stored in procedural fashion." StAX was created to address limitations in the two most prevalent parsing APIs, SAX and DOM.

The StAX API exposes methods for iterative, event-based processing of XML documents. XML documents are treated as a filtered series of events, and infoset states can be stored in a procedural fashion. Moreover, unlike SAX, the StAX API is bidirectional, enabling both reading and writing of XML documents.

The StAX API is really two distinct API sets: a **cursor** API and an **iterator** API.

Cursor API

As the name implies, the StAX **cursor** API represents a cursor with which you can walk an XML document from beginning to end. This cursor can point to one thing at a time, and always moves forward, never backward, usually one infoset element at a time. The two main cursor interfaces are XMLStreamReader and XMLStreamWriter. XMLStreamReader includes accessor methods for all possible information retrievable from the XML Information model, including document encoding, element names, attributes, namespaces, text nodes, start tags, comments, processing instructions, document boundaries, and so forth.

Iterator API

The StAX *iterator* API represents an XML document stream as a set of discrete event objects. These events are pulled by the application and provided by the parser in the order in which they are read in the source XML document.

The base iterator interface is called XMLEvent, and there are subinterfaces for each event type listed in Table 3-2, below. The primary parser interface for reading iterator events is XMLEventReader, and the primary interface for writing iterator events is XMLEventWriter. The XMLEventReader interface contains five methods, the most important of which is nextEvent(), which returns the next event in an XML stream. XMLEventReader implements java.util.Iterator, which means that returns from XMLEventReader can be cached or passed into routines that can work with the standard Java Iterator

Referenced links

http://download.oracle.com/javaee/5/tutorial/doc/bnbdv.html

Sample

http://download.oracle.com/javaee/5/tutorial/doc/bnbfl.html