**Читаем XML файл в Java с помощью StAX**

В этой статье мы продолжим изучать различные API по обработке XML в Java. Сегодня мы разберем Java StAX API: разберем наборы API, которые предоставляет StAX и попробуем на практике разобрать простой XML  файл.

**Теоретические сведения о Java StAX API**

Java StAX или Java Streaming API для XML предоставляет API для обработки XML в Java. StAX состоит из двух наборов API: на основе итератора и на основе курсора.

* API на основе курсора позволяет приложению обрабатывать XML как поток событий: приложение может проверить состояние синтаксического анализатора и получить информацию о последнем анализируемом событии, а затем перейти к следующему событию, и так далее. **Это низкоуровневый API, который не обеспечивает абстракцию над XML структурой**.
* **Второй набор — высокоуровневый API на основе итератора**. Этот API позволяет приложению обрабатывать XML как серию объектов-событий, каждый из которых представляет часть структуры XML. Для каждого разобранного события нужно определить тип, а также разобрать событие на составляющие блоки.

**Используем Java StAX API для разбора XML файла**

**Java StAX** обеспечивает фабричные методы для создания XMLEventReader с помощью которого мы можем представить любой элемент XML файла в виде XMLEvent.

Наиболее полезными методами класса XMLEvent являются isStartElement() и метод isEndElement(), которые отвечают за определение начального  и конечного тега.

Ниже представлен XML, на который мы будем натравливать Java StAX:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <?xml version="1.0" encoding="UTF-8"?>  <Students>      <Student id="1">          <name>Andrey</name>          <age>21</age>          <language>Java</language>      </Student>      <Student id="2">          <name>Viktor</name>          <age>20</age>          <language>C++</language>      </Student>  </Students> |

Ниже представлен листинг класса XMLstaxParser для чтения XML файла:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76 | package ua.com.prologistic;    import java.io.FileInputStream;  import java.io.FileNotFoundException;  import java.util.ArrayList;  import java.util.List;    import javax.xml.namespace.QName;  import javax.xml.stream.XMLEventReader;  import javax.xml.stream.XMLInputFactory;  import javax.xml.stream.XMLStreamException;  import javax.xml.stream.events.Attribute;  import javax.xml.stream.events.EndElement;  import javax.xml.stream.events.StartElement;  import javax.xml.stream.events.XMLEvent;    public class XMLstaxParser {        public static void main(String[] args) {          String fileName = "F:/students.xml";          List<Student> studentsList = parseXMLfile(fileName);          // печатаем в консоль информацию по каждому студенту          for (Student student : studentsList) {              System.out.println(                      student.toString());          }      }        private static List<Student> parseXMLfile(String fileName) {          List<Student> studentsList = new ArrayList<>();          Student student = null;          XMLInputFactory xmlInputFactory = XMLInputFactory.newInstance();          try {              // инициализируем reader и скармливаем ему xml файл              XMLEventReader reader = xmlInputFactory.createXMLEventReader(new FileInputStream(fileName));              // проходим по всем элементам xml файла              while (reader.hasNext()) {                  // получаем событие (элемент) и разбираем его по атрибутам                  XMLEvent xmlEvent = reader.nextEvent();                  if (xmlEvent.isStartElement()) {                      StartElement startElement = xmlEvent.asStartElement();                      if (startElement.getName().getLocalPart().equals("Student")) {                          student = new Student();                          // Получаем атрибут id для каждого элемента Student                          Attribute idAttr = startElement.getAttributeByName(new QName("id"));                          if (idAttr != null) {                              student.setId(Integer.parseInt(idAttr.getValue()));                          }                      } else if (startElement.getName().getLocalPart().equals("age")) {                          xmlEvent = reader.nextEvent();                          student.setAge(Integer.parseInt(xmlEvent.asCharacters().getData()));                      } else if (startElement.getName().getLocalPart().equals("name")) {                          xmlEvent = reader.nextEvent();                          student.setName(xmlEvent.asCharacters().getData());                      } else if (startElement.getName().getLocalPart().equals("language")) {                          xmlEvent = reader.nextEvent();                          student.setLanguage(xmlEvent.asCharacters().getData());                      }                  }                  // если цикл дошел до закрывающего элемента Student,                  // то добавляем считанного из файла студента в список                  if (xmlEvent.isEndElement()) {                      EndElement endElement = xmlEvent.asEndElement();                      if (endElement.getName().getLocalPart().equals("Student")) {                          studentsList.add(student);                      }                  }              }            } catch (FileNotFoundException | XMLStreamException exc) {              exc.printStackTrace();          }          return studentsList;      }    } |

Как видно из листинга, мы разбираем XML файл в список объектов Student с помощью метода parseXMLfile(). Результат работы программы представлен ниже:

|  |  |
| --- | --- |
| 1  2 | Student{name='Andrey', age=21, id=1, language='Java'}  Student{name='Viktor', age=20, id=2, language='C++'} |

Кому интересен способ чтения файла с помощью DOM Parser, перейдите по [этой ссылке](http://prologistic.com.ua/kak-chitat-xml-fajl-v-java-ispol-zuem-dom-parser.html).