Java StAX Parser - Query XML Document

Java StAX parser is a Java API which is used to parse XML documents and query the necessary information. This API is event based and hence we need not load entire XML document to query it. As the parser identifies each event, the corresponding action is performed only when the client program implements the event.

In this chapter, we are going to see how to query an XML document to get necessary information.

Query XML Using Java StAX Parser

Following are the steps we need to follow to query an XML document using Java StAX parser –

- **Step 1:** Creating XMLInputFactory instance
- **Step 2:** Reading the XML
- **Step 3:** Parsing the XML
- **Step 4:** Querying the Elements

Refer this chapter for first three steps

Step 4: Querying the Elements

After following the first three steps, we have XMLEventReader to get events from the XML document. Using various events and by implementing them, we can query the XML document. Let us see how we can do this in detail.

Querying Elements by Text Content

We can get the text content in any element of an XML document by using **getData()**.This method of Characters interface returns the character data of the current event in the form of a String. Using this method, we can query all the elements to find the required text content.

Example

The **cars.xml** file shown below has many carname elements with different text content. Let us query this file to find out if "Bentley 2" car is present.

```
<?xml version = "1.0"?>
<cars>
      <carname company="Ferarri" >Ferarri 101</carname>
      <carname company="Lamborgini">Lamborgini 001</carname>
      <carname company="Lamborgini">Lamborgini 002</carname>
      <carname company="Lamborgini">Lamborgini 003</carname>
      <carname company="Bentley">Bentley 1</carname>
      <carname company="Bentley">Bentley 2</carname>
      <carname company="Bentley">Bentley 3</carname>
</cars>
```

The following **QueryTextContent.java** program reads the cars.xml file using a FileReader object. When the CHARACTERS event type is encountered, the getData() method is used to get the text content. If that data is equal to "Bentley 2", then we are updating the 'found' boolean variable. In the END_DOCUMENT event, we are printing it on the console.

```
import java.io.FileReader;
import javax.xml.stream.XMLEventReader;
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamConstants;
import javax.xml.stream.events.Characters;
import javax.xml.stream.events.XMLEvent;

public class QueryTextContent {
   public static void main(String args[]) {
      try {

         //Creating XMLInputFactory instance
         XMLInputFactory factory = XMLInputFactory.newInstance();

         //Reading the XML
         FileReader fileReader = new FileReader("cars.xml");

         //Parsing the XML
         XMLEventReader eventReader =
          factory.createXMLEventReader(fileReader);

         //Querying the XML
         boolean found=false;
         while(eventReader.hasNext()) {
```

```
            XMLEvent event = eventReader.nextEvent();
            if(event.getEventType()==XMLStreamConstants.CHARACTERS) {
             Characters characters = event.asCharacters();
             String textContent = characters.getData();
             if(textContent.equals("Bentley 2"))
                 found=true;
            }
            if(event.getEventType()==XMLStreamConstants.END_DOCUMENT) {
                if(found)
                   System.out.println("Bentley 2 car is found");
                else
                   System.out.println("Bentley 2 car is not found");
            }
         }

      } catch(Exception e) {
           e.printStackTrace();
      }
   }
}
```

Output

Since, Bentley 2 car is present in cars.xml file, it prints that it is found.

```
Bentley 2 car is found
```

Learn **Java** in-depth with real-world projects through our **Java certification course**. Enroll and become a certified expert to boost your career.

Querying Elements by Attributes

The **getAttributeByName()** method of an Element interface takes the QName object which is the qualified XML name of an attribute and returns the Attribute object. Further, the **getValue()** method of Attribute interface is used to get the value of the attribute in the form of a String.

Example 1

The cars.xml file that we have used in the previous example is parsed in the following **QueryAttributes.java** program to count the number of Bentley cars present in the XML file. A count variable is incremented each time the company attribute of the carname element is equal to Bentley.

```
import java.io.FileReader;
import javax.xml.namespace.QName;
import javax.xml.stream.XMLEventReader;
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamConstants;
import javax.xml.stream.events.Attribute;
import javax.xml.stream.events.StartElement;
import javax.xml.stream.events.XMLEvent;

public class QueryExample2 {
   public static void main(String args[]) {
      try {

         //Creating XMLInputFactory instance
         XMLInputFactory factory = XMLInputFactory.newInstance();

         //Reading the XML
         FileReader fileReader = new FileReader("cars.xml");

         //Parsing the XML
         XMLEventReader eventReader =
           factory.createXMLEventReader(fileReader);

          //Querying the XML document
          int count=0;
          while(eventReader.hasNext()) {
              XMLEvent event = eventReader.nextEvent();
              if(event.getEventType()==XMLStreamConstants.START_ELEMENT) {
                StartElement element=event.asStartElement();
                QName qname=new QName("company");
                Attribute attr=element.getAttributeByName(qname);
                if(attr!=null && attr.getValue().equals("Bentley"))
                   count++;
              }
          }
          System.out.println("No.of Bentley cars found : " + count);
      } catch(Exception e) {
           e.printStackTrace();
      }
   }
}
```

The number of Bentley cars in the XML file are displayed.

```
No.of Bentley cars found : 3
```

Example 2

In this example, we are going to retrieve the information of a particular student based on their roll number. Here is the **student.xml** file we need to query −

```
<?xml version = "1.0"?>
<class>
   <student rollno = "393">
      <firstname>dinkar</firstname>
      <lastname>kad</lastname>
      <nickname>dinkar</nickname>
      <marks>85</marks>
   </student>

   <student rollno = "493">
      <firstname>Vaneet</firstname>
      <lastname>Gupta</lastname>
      <nickname>vinni</nickname>
      <marks>95</marks>
   </student>

   <student rollno = "593">
      <firstname>jasvir</firstname>
      <lastname>singn</lastname>
      <nickname>jazz</nickname>
      <marks>90</marks>
   </student>
</class>
```

In the following **QueryStudent.java** program, we have checked for the events START_ELEMENT, CHARACTERS and END_ELEMENT and performed actions accordingly. when the roll number attribute is equal to 393, we are printing the entire information of the student.

```
import java.io.FileReader;
import java.util.Iterator;
import javax.xml.stream.XMLEventReader;
```

```
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamConstants;
import javax.xml.stream.events.Attribute;
import javax.xml.stream.events.Characters;
import javax.xml.stream.events.EndElement;
import javax.xml.stream.events.StartElement;
import javax.xml.stream.events.XMLEvent;

public class QueryStudent {
   public static void main(String[] args) {
      boolean bFirstName = false;
      boolean bLastName = false;
      boolean bNickName = false;
      boolean bMarks = false;
      boolean isRequestRollNo = false;

      try {

         //Creating XMLInputFactory instance
         XMLInputFactory factory = XMLInputFactory.newInstance();

         //Reading the XML
         FileReader fileReader = new FileReader("student.xml");

         //Parsing the XML
         XMLEventReader eventReader =
            factory.createXMLEventReader(fileReader);

         //Querying the XML
         String requestedRollNo = "393";
         while(eventReader.hasNext()) {
            XMLEvent event = eventReader.nextEvent();

            switch(event.getEventType()) {
               case XMLStreamConstants.START_ELEMENT:
                  StartElement startElement = event.asStartElement();
                  String qName = startElement.getName().getLocalPart();

               if (qName.equalsIgnoreCase("student")) {
                  Iterator<Attribute> attributes =
startElement.getAttributes();
                  String rollNo = attributes.next().getValue();
```

```
            if(rollNo.equalsIgnoreCase(requestedRollNo)) {
               System.out.println("Start Element : student");
               System.out.println("Roll No : " + rollNo);
               isRequestRollNo = true;
            }
         } else if (qName.equalsIgnoreCase("firstname")) {
            bFirstName = true;
         } else if (qName.equalsIgnoreCase("lastname")) {
            bLastName = true;
         } else if (qName.equalsIgnoreCase("nickname")) {
            bNickName = true;
         }
         else if (qName.equalsIgnoreCase("marks")) {
            bMarks = true;
         }
         break;

         case XMLStreamConstants.CHARACTERS:
            Characters characters = event.asCharacters();

         if(bFirstName && isRequestRollNo) {
            System.out.println("First Name: " + characters.getData());
            bFirstName = false;
         }
         if(bLastName && isRequestRollNo) {
            System.out.println("Last Name: " + characters.getData());
            bLastName = false;
         }
         if(bNickName && isRequestRollNo) {
            System.out.println("Nick Name: " + characters.getData());
            bNickName = false;
         }
         if(bMarks && isRequestRollNo) {
            System.out.println("Marks: " + characters.getData());
            bMarks = false;
         }
         break;

         case XMLStreamConstants.END_ELEMENT:
            EndElement endElement = event.asEndElement();
```

```
                if(endElement.getName().getLocalPart().equalsIgnoreCase(
                   "student") && isRequestRollNo) {
                   System.out.println("End Element : student");
                   System.out.println();
                   isRequestRollNo = false;
                }
                break;
             }
          }
       } catch (Exception e) {
          e.printStackTrace();
       }
    }
}
```

All the information of the student with roll number 393 is displayed.

```
Start Element : student
Roll No : 393
First Name: dinkar
Last Name: kad
Nick Name: dinkar
Marks: 85
End Element : student
```