

Java SAX Parser - Query XML Document

Java SAX Parser is an API in java which can be used to query XML documents. To query large XML documents using event based APIs, we can use Java SAX parser API. The SAXParser API is present inside the javax.xml.parsers package. We can query any XML document by implementing the call back methods inside Handler class by inheriting DefaultHandler class which in turn inherits ContentHandler interface.

Query XML using Java SAX parser

We can query any XML document in Java using SAX parser through following steps –

- **Step 1:** Implementing a Handler class
- **Step 2:** Creating a SAXParser Object
- **Step 3:** Reading the XML
- **Step 4:** Creating object for Handler class
- **Step 5:** Parsing the XML Document
- **Step 6:** Querying the Elements

Refer [this chapter](#) for the first five steps.

Step 6: Querying the Elements

The ContentHandler and Attributes interfaces provide various methods which help us query elements and their attributes. For querying required information, We implement code inside our Handler class.

Querying Elements by TextContent

The **characters()** method is used to get the content of elements. The **endDocument()** function is called at the end of the document. We can use this method to perform some necessary actions that we need to do before closing our XML file.

Example

We need to query the following **cars.xml** file to find out whether there is "Bentley 2" car in the document.

```
<?xml version = "1.0"?>
<cars>
```

```
<carname company="Ferrari" >Ferrari 101</carname>
<carname company="Lamborghini">Lamborghini 001</carname>
<carname company="Lamborghini">Lamborghini 002</carname>
<carname company="Lamborghini">Lamborghini 003</carname>
<carname company="Bentley">Bentley 1</carname>
<carname company="Bentley">Bentley 2</carname>
<carname company="Bentley">Bentley 3</carname>
</cars>
```

In the following java program, we have implemented characters() and endDocument() methods inside UserHandler class. We have written the code to find Bentley cars inside the characters() function and the end result is printed using endDocument() method.

```
import java.io.File;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

//Implementing UserHandler Class
class UserHandler extends DefaultHandler {

    boolean found = false;

    public void characters(char[] ch, int start, int length) throws
SAXException{
        String textContent = new String(ch,start,length);
        if(textContent.equals("Bentley 2")) {
            found=true;
        }
    }
    public void endDocument() {
        if(found)
            System.out.println("Bentley 2 car is found");
        else
            System.out.println("Bentley 2 car is not Found");
    }
}

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

```
//Creating a DocumentBuilder Object
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParser saxParser = factory.newSAXParser();

//Reading the XML
File xmlFile = new File("cars.xml");

//Creating UserHandler object
UserHandler userHandler = new UserHandler();

//Parsing the XML Document
saxParser.parse(xmlFile, userHandler);

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

The output displays that the Bentley car 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

We can query the elements by their attributes using the Attributes interface. We get the list of attributes of an element as a last argument from the startElement() function. As discussed in the previous chapter, the **getValue("attr_name")** function is used to obtain the attribute value.

Example

Let us use our cars.xml file that we used in the previous example to count number of Bentley cars. This information can be obtained only through the attribute value. As the parser encounters an element, we are using getValue("company") to get the company attribute and incrementing the count if that company is Bentley.

```
import java.io.File;
import javax.xml.parsers.SAXParser;
```

```
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

//Implementing UserHandler Class
class UserHandler extends DefaultHandler {

    int count=0;

    public void startElement(String uri, String localName, String qName,
Attributes attributes) throws SAXException {
        String company = attributes.getValue("company");
        if(qName.equals("carname") && company.equals("Bentley")) {
            count++;
        }
    }
    public void endDocument() {
        System.out.println("No. of Bentley cars found : " + count);
    }
}

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

            //Creating a DocumentBuilder Object
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();

            //Reading the XML
            File xmlFile = new File("cars.xml");

            //Creating UserHandler object
            UserHandler userHandler = new UserHandler();

            //Parsing the XML Document
            saxParser.parse(xmlFile, userHandler);

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

```
}  
}
```

The output is displayed when the endDocument() method is invoked.

No.of Bentley cars found : 3