第十次实验：使用 JDK 解析 XML 文件

段 炼

2010 年 6 月 6 日

Java 中进行 DOM 的 XML 编程不需要其它的依赖包，因为 JDK 里自带的 org.w3c.dom、org.xml.sax 和 javax.xml.parsers 包就可以完成任务。

（1）org.w3c.dom　W3C 推荐的用于 XML 标准规划文档对象模型的接口。

（2）org.xml.sax　用于对 XML 进行语法分析的事件驱动的 XML 简单 API（SAX）

（3）javax.xml.parsers 解析器工厂工具，程序员获得并配置特殊的特殊语法分析器。

要解析的 XML 文件：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE book SYSTEM "D:\workspace\XML\WebRoot\WEB-INF\book.dtd">
<book>
    <bookname name="WebGIS" font="GB2312"></bookname>
    <authors>
        <author name="Tom" sex="male" age="45"></author>
        <author name="lihongbo" sex="male" age="35"></author>
        <author name="wangxiaoer “ sex="male" age="30"></author>
    </authors>
    <price value="85"></price>
    <publishdate>
        <value>2006-08-18</value>
    </publishdate>
</book>
```

解析 XML 文件的 Java 文件：

package xml;

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class XPathForXml {
        public void parseXMLWithJdk(){
            try {

                //读取 book.xml 到内存

                DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
                DocumentBuilder dbd = dbf.newDocumentBuilder();
                Document doc = dbd.parse(new FileInputStream("D:\\book.xml"));


                //通过 XML 获得 book 的 authors 的 author 子节点列表

                XPathFactory f = XPathFactory.newInstance();
                XPath path = f.newXPath();
                NodeList authors= (NodeList) path.evaluate("book/authors/author",
doc,XPathConstants.NODESET);
                System.out.println(authors.getLength());

                //遍历取到的元素

                if(authors!=null){
                    for(int i=0;i<authors.getLength();i++){
                     Node author      = authors.item(i);
                     int n = i + 1;

                     System.out.print(n+".  名字:"+author.getNodeName());

                     System.out.println();
                    }
```

```
            }

            //获得 book 的 authors 的第一个子节点,注意 NODESET 和 NODE 的区别
            Node author= (Node) path.evaluate("book/authors/author",
doc,XPathConstants.NODE);
            System.out.println("  名称："+author.getNodeName());
            System.out.println("  内容："+author.getTextContent());//如果存在内容则返回内容，
不存在则返回空
            //获取节点的属性
            NamedNodeMap attr =  author.getAttributes();
            System.out.println("  该节点的属性个数"+attr.getLength());
            //遍历元素的属性
            if(attr!=null){
               for(int i=0;i<attr.getLength();i++){
                  int n = i + 1;
                  System.out.print("  属性"+n+"  名称:"+attr.item(i).getNodeName());
                  System.out.print("  值:"+attr.item(i).getNodue());
                  System.out.print("  类型:"+attr.item(i).getNodeType());
                  System.out.println();
               }
            }

         } catch (ParserConfigurationException e) {
            e.printStackTrace();
         } catch (FileNotFoundException e) {
            e.printStackTrace();
         } catch (SAXException e) {
            e.printStackTrace();
         } catch (IOException e) {
            e.printStackTrace();
         } catch (XPathExpressionException e) {
            e.printStackTrace();
         }
      }
```

```
    public static void main(String[] args) {
        new XPathForXml().parseXMLWithJdk();
    }
}
```

代码讲解：

（1）得到 DOM 解析器的工厂实例

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

得到 javax.xml.parsers.DocumentBuilderFactory;类的实例就是我们要的解析器工厂

（2）从 DOM 工厂获得 DOM 解析器

DocumentBuilder dbd = domfac.newDocumentBuilder();

通过 javax.xml.parsers.DocumentBuilderFactory 实例的静态方法 newDocumentBuilder()得到

DOM 解析器

（3）把要解析的 XML 文档转化为输入流，以便 DOM 解析器解析它

InputStream is=new FileInputStream("D:\\book.xml");

InputStream 是一个接口。

（4）解析 XML 文档的输入流，得到一个 Document

Document doc=dombuilder.parse(is);

由 XML 文档的输入流得到一个 org.w3c.dom.Document 对象，以后的处理都是对 Document

对象进行的