

J2EE中XML配置文件的读取处理 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_J2EE_E4_B8_ADXML_c104_144728.htm Java和XML是黄金组合，网上已经有很多文章介绍，XML作为电子商务中数据交换，已经有其不可替代的作用，但是在平时系统开发中，我们不一定都用到数据交换，是不是无法使用XML了？当然不是，现在已经有一个新趋势，Java程序的配置文件都开始使用XML格式，以前是使用类似windows的INI格式(Java中也有Propertiesy这样的类专门处理这样的属性配置文件)。使用XML作为Java的配置文件有很多好处，从Tomcat的安装配置文件和J2ee的配置文件中，我们已经看到XML的普遍应用，让我们也跟随流行趋势用XML武装起来。现在关键是如何读取XML配置文件？有好几种XML解析器：主要有DOM和SAX，在apache的XML项目组中，目前有Xerces Xalan Cocoon几个开发XML相关技术的project。Tomcat本身使用的是Sun的JAXP，而其XSL Taglib project中使用Xerces解析器。好了，上面都是比较烦人的理论问题，还是赶快切入XML的配置文件的读取吧。在我们的程序中，通常要有一些根据主机环境确定的变量。比如数据库访问用户名和密码，不同的主机可能设置不一样。只要更改XML配置文件，就可以正常运行。

localhostsqlnameusernamepassword 上面这个myenv.xml配置文件一般是放在tomcat的WEB-INF/classes目录下。我们编制一个Java程序直接读取，将dbhost dbuser dbpassword提取出来供其他程序访问数据库用。目前使用SAX比较的多，与DOM主要区别是SAX是一行一行读取XML文件进行分析，适合比较

大文件，DOM是一次性读入内存，显然不能对付大文件。这里我们使用SAX解析，由于SAX解析器不断在发展，网上有不少文章是针对老版本的。如果你使用JDK1.4，可以参考使用SAX处理XML文档一文。这里的程序是根据其改进并且经过实践调试得来的。来源：www.examda.com 对上

面myenv.xml读取的Java程序:

```
import
```

```
org.xml.sax.Attributes.import
```

```
org.xml.sax.helpers.DefaultHandler.import
```

```
org.xml.sax.SAXException. import java.util.Properties.//使
```

```
用DefaultHandler的好处是不必陈列出所有方法， public class  
ConfigParser extends DefaultHandler{////定义一个Properties 用来  
存放dbhost dbuser dbpassword的值private Properties props.private
```

```
String currentSet.private String currentName.private StringBuffer
```

```
currentValue = new StringBuffer().//构建器初始化propspublic
```

```
ConfigParser(){this.props = new Properties().}public Properties
```

```
getProps(){return this.props.}//定义开始解析元素的方法。 这里
```

```
是将中的名称xxx提取出来。 public void startElement(String uri,
```

```
String localName,String qName, Attributes attributes) throws
```

```
SAXException{currentValue.delete(0,
```

```
currentValue.length()).this.currentName =qName.}//这里是将之
```

```
间的值加入到currentValuepublic void characters(char[] ch,int
```

```
start, int length) throws SAXException{ currentValue.append(ch,
```

```
start, length).}//在遇到结束后，将之前的名称和值一一对应保
```

```
存在props中public void endElement(String uri,String localName,
```

```
String qName)throws SAXException
```

```
{props.put(qName.toLowerCase(),
```

currentValue.toString().trim()).}} 上面的这个解析程序比较简单吧? 其实解析XML就是这么简单。现在我们已经将dbhost dbuser dbpassword的值localhost sqlname username password提取了出来。但是这只是在在解析器内部, 我们的程序还不能访问。需要再编制一个程序。

```
import java.util.Properties;import
javax.xml.parsers.SAXParser;import
javax.xml.parsers.SAXParserFactory;import java.net.URL;public
class ParseXML{//定义一个Properties 用来存放 dbhost dbuser
dbpassword的值private Properties props;//这里的propspublic
Properties getProps(){return this.props;}public void parse(String
filename)throws Exception {//将我们的解析器对象
化ConfigParser handler = new ConfigParser();//获取SAX工厂对
象SAXParserFactory factory=
SAXParserFactory.newInstance().factory.setNamespaceAware(false)
.factory.setValidating(false);//获取SAX解析SAXParser parser=
factory.newSAXParser();//得到配置文件myenv.xml所在目录
。tomcat中是在WEB-INF/classes//下例中BeansConstants是用来
存放xml文件中配置信息的类, 可以自己代替或定义URL
confURL =
BeansConstants.class.getClassLoader().getResource(filename).try{//
将解析器和解析对象myenv.xml联系起来, 开始解析
parser.parse(confURL.toString(), handler). //获取解析成功后的属
性 以后我们其他应用程序只要调用本程序的props就可以提取
出属性名称和值了props =
handler.getProps().}finally{factory=null.parser=null.handler=null.}}
}
```

由于我们的XML文件是使用最简单的形式, 因此解析器相

对简单，但是这已经足够对付我们的配置文件了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com