

使用JDOM对XML文件进行操作的实例讲解 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/237/2021\\_2022\\_\\_E4\\_BD\\_BF\\_E7\\_94\\_A8JDOM\\_c104\\_237355.htm](https://www.100test.com/kao_ti2020/237/2021_2022__E4_BD_BF_E7_94_A8JDOM_c104_237355.htm)

本文着重介绍在应用程序中如何使用JDOM对XML文件进行操作，要求读者具有基本的JAVA语言基础。XML由于其可移植性，已经成为应用开发中必不可少的环节。我们经常会把应用程序的一些配置文件（属性文件）写成XML的格式（当然，也可以用property文件而不用XML文件），应用程序通过XML的访问类来对其进行操作。对XML进行操作可以通过若干种方法，如：SAX, DOM, JDOM, JAXP等，JDOM由于其比较简单实用而被开发人员普遍使用。本文主要分两部分，第一部分介绍如何把XML文件中的配置读入应用程序中，第二部分介绍如何使用JDOM将配置输出到XML文件中。以下是一段XML配置文件，文件名为contents.xml：

```
Java and XML XML Matters Whats Important The Essentials Whatamp.apos.s Next? 下面的程序通过使用JDOM中SAXBuilder类对contents.xml进行访问操作，把各个元素显示在输出console上，程序名为：SAXBuilderTest.java，内容如下：
```

```
import java.io.File;import java.util.Iterator;import java.util.List;import org.jdom.Document;import org.jdom.Element;import org.jdom.input.SAXBuilder;public class SAXBuilderTest { private static String titlename. private String chapter. private String topic. public static void main(String[] args) { try { SAXBuilder builder = new SAXBuilder(). Document document = builder.build(new File("contents.xml")). Element root = document.getRootElement(). Element title = root.getChild("title").
```

```
titlename = title.getText(). System.out.println("BookTitle: "
titlename). Element contents = root.getChild("contents"). List
chapters = contents.getChildren("chapter"). Iterator it =
chapters.iterator(). while (it.hasNext()) { Element chapter =
(Element) it.next(). String chaptertitle =
chapter.getAttributeValue("title"). System.out.println("ChapterTitle:
" chaptertitle). List topics = chapter.getChildren("topic"). Iterator
iterator = topics.iterator(). while (iterator.hasNext()) { Element topic
= (Element) iterator.next(). String topicname = topic.getText().
System.out.println("TopicName: " topicname). } } } catch
(Exception ex) { } } } 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
```