使用 Java 编程解析 Web 页面

Parsing Web Page Document with Java Programming

刘遵雄（西安交通大学网络所 710049、南昌华东交通大学电气学院 330013）
聂国星（南昌华东交通大学网络所 710049）

摘要：本文研究并探讨了使用 Java 的 Swing 包编程解析 Web 页的关键技术，并提供了具体的解决实例和实现中以及要注意的问题，实现了 web 信息的提取。

关键词：Java 编程 HTML 文件 解析技术

1 引言

在当今网络环境下，解析 HTML 文档的工作时刻在进行着。我们总是使用某个搜索引擎在互联网这个巨量信息库里检索信息，如 Google 和 Yahoo 等，提高了工作效率。我们之所以能够享受搜索引擎带来的方便快捷，是与一种可称为网络蜘蛛(Spider)的软件使用分不开的，运行的 Spider 不断在网上通过链接巡游，并且对访问的 HTML 文档进行解析记录。目前市场上存在一些 HTML 解析器，有时它们不能很好地满足各自要求，这就要求用户自己进行编程对 HTML 文档进行解析。Java 语言作为一种流行的网络编程工具，在一定程度上提供解析 HTML 文档的功能。本文以下两部分就使用 Java 进行 HTML 解析的主要技术和具体实现过程的主要环节进行了探讨和研究，并给予相应的重点提示。

2 主要技术

2.1 使用 HTMLEditorKit.Parser

主要的 HTML 解析类 Parser 是 HTMLEditorKit 类的内部类，包含在 Swing 包中。其实一个抽象类，使用它进行工作关键在于将类 Parser 进行实例化。Java 对类 Parser 实行如此包装并没用考虑到外部解析 HTML 的功能需求，似乎产生了一些负面影响，我们可以使用 HTMLEditorKit.Parser 类实例化的方法，这种唯一的方法是通过重载 HTMLEditorKit 类的 getParser 方法来实例化 HTMLEditorKit.Parser 对象，从而实现其 public 访问性。实现访问 Swing 的 HTML 解析器的类 HTMLParser 定义文件 HTMLParse.java 如下：

```
Import javax.swing.text.html.*;
Public class HTMLParse extends
HTMLEditorKit
{ public HTMLEditorKit.Parser getParser()
  {
    //返回新的 HTMLEditorKit.Parser 对象 return super.getParser();
  }}
```

这个类的实例从 Reader 读取 HTML 文档，并在文档中寻找五种标签：开始标签、结束标签、空标签、文本和注释，它们涵盖了普通 HTML 文件的所有重要部分。每次解析器看到这五个标签之一时，它就会调用相应的回调方法，回调方法在 javax.swing.text.html.HTMLEditorKit.ParserCallback 类中的定义。为了解析一个 HTML 文件，用户写一个回应文本和标签的 HTMLEditorKit.ParserCallback 子类，然后传递一个用户子类的实例到 HTMLEditorKit.Parser 的 parse 方法中，待处理的 HTML 文件的 Reader 对象将一同作为参数传递，parse 方法的第三个参数用于指定是否注意文档的字符集，一般取 true。

```
StringReader r=new
StringReader(…html string…);
HTMLEditorKit.Parser parse=new
HTMLParse().getParser();
Parser.parse(r,callback.true);
```

通过调用 HTMLEditorKit 的 getParser 方法来实现 Parser 对象的实例化，该方法没有公共访问接口，在子类中重载 getParser 使之成为公共成员函数。获得了 Parser 类，就可以调用其 parse 方法，parse 方法只是 HTMLEditorKit.Parser 类的公用方法，所有这些都是在 HTMLEditorKit.ParserCallback 子类的回调方法内部处理的。Parse() 只是简单地从 Reader 中读取数据直到读完整个文档，每次看到标签、注释或者文本，都会调用 HTMLEditorKit.ParserCallback 实例的相关回调方法。HTMLEditorKit.Parser 和 HTMLEditorKit.ParserCallback 实例都是独立阅读器的，

可以解析多个文件,只是调用 Parse()多次。对于 HTML 数据流中的每个种类的标签(tag)都会反复调用 callback 对象。以下讨论 ParserCallback 类的结构。

2.2 使用 HTMLEditorKit.ParserCallback

ParserCallback 是 HTMLEditorKit 内的公用内部类,必须使用 ParserCallback 类创建将用到的解析器,构建 ParserCallback 类的子类就是实现对待处理的文件进行解析,通过重载 handleComment, handleEndTag, handleError, handleSimpleTag, handleStartTag, 和 handleText? 方法,在文档解析时响应输入流中的具体项目:

Public void handleComment (char[] data ,int pos)

Public void handleEndTag(HTML.Tag t,int pos)

Public void handleError(String errorMsg,int pos)

Public void handleSimpleTag(HTML.Tag t,MutableAttributeSet a,int pos)

Public void handleStartTag(HTML.Tag t,MutableAttributeSet a,int pos)

Public void handleText(char[] data,int pos)

就几个我们将使用到的回调方法说明如下:

Public void handleSimpleTag(HTML.Tag t,MutableAttributeSet a,int pos)

当处理到 HTML 文档中的简单标签时,解析器调用 handleSimpleTag 方法,参数 t 指代定位到的标签,a 包含有属性列表,参数 pos 指示当前位置。简单标签有时也是结束标签,解析器解析到那种被认为是简单或未知标签的结尾标签时,将调用 handleSimpleTag 方法。判别标签是否为结尾标签或真正的简单标签的唯一途径,是检查标签中是否存在 HTML.Attribute.ENDTAG 属性,只有结尾标签才包含该属性。

Public void handleStartTag(HTML.Tag t,MutableAttributeSet a,int pos)

当处理到 HTML 文档中的开始标签时,解析器调用 handleStartTag? 方法,参数 t 指代定位到的 a 包含有属性列表,参数 pos 指示当前位置。Public void handleText (char[]data,int pos)

当处理到 HTML 文档中的文本(TEXT,包含 HTML 文档中的任何显示文本)时,解析器调用 handleTex 方法。其中参数 data 中包含了当前的文本,参数 pos 指示当前的位置。例如 data 字符数组中数据为"Yahoo",那么 pos=2 时 data[pos]对应的字符为"h"。

2.3 使用标签(HTML.Tag)

HTML.Tag 类是公共内部类,用来表示 HTML 标签,将被传递到 ParserCallback 类提供的回调方法中。HTML.Tag 类提供四种方法来返回关于标签的基本信息,它们是:

Public Boolean breaksFlow()若标签产生一个分行标志,则返回 true。

Public Boolean isBlock()若标签产生一个分段标志,则返回 true。

Public Boolean isPreformatted()若标签表示应当保留的空格,则返回 true。

Public String toString()将标签名转换为字符串,以便处理。

用户通常只是想了解标签的类型,为了达到此目的,可以使用 HTML.Tag 中预定义的标签常量来处理,包含了 75 个能为 HTMLEditorKit.Parser 识别的标签常量,通过 HTMLEditorKit.Parser 传递到回调方法的 HTML.Tag 元素,都是这 75 个对象中的一个。例如:判别变量 t 中的标签是否为 H1 标签,使用的代码为:if(t==HTML.Tag.H1)

.....

2.4 使用属性(HTML.Attribute)

特定标签具有不同的属性,处理 HTML 文件离不开对标签属性的了解。HandleSimpleTag() 和 handleStartTag()回调方法的第二个参数是: javax.swing.text.MutableAttributeSet 类的一个实例,这个对象允许用户察看什么属性依附在指定标签上,代表了 HTML 标签上的属性集合。HTML.Attribute 类定义了 80 个能为 HTMLEditorKit.Parser 识别的标准属性常量。(HTML.Attribute.ACTION, HTML.Attribute.ALIGN 和 HTML.Attribute.HREF 等等)。

MutableAttributeSet 类是 javax.swing.text.AttributeSet 接口的子接口,这些接口声明了相关的方法,详情参看有关资料。例如:检索 MutableAttributeSet 对象 a 中的 HREF 属性,使用的代码是:

String value = a.getAttribute (HTML.Attribute.HREF)

3 实现问题

本文实现了读入一个网页 HTML 文件的 URL 地址,如 http://www.hao123.com/campus.htm,或者一个本地 HTML 文件地址,如"file:/d:/Example/Page1.html",通过使用以上研究过来的技术将网页中的超级链接名和相应的地址解析出来,分别对应存入两个 Collection 对象 workname 和 workurl 容器中。其中要特别注意在 HTMLEditorKit.Parser 使用 parse()方法调用 HTMLEditorKit.ParserCallback 类对象的回调方法处理标记、文本等逻辑顺序和超级链接名与其 URL 一一对应的问题。

3.1 HTML文件的读取

要成功地实现对网页 HTML 文件的解析，必须文档通过输入流读入阅读器对象 Reader 中，首先通过抽象类 URLConnection 建立与 URL 指定的数据源的动态链接。再实现 HTMLEditorKit. Parser 类和 HTMLEditorKit. ParserCallBack 类的实例化，最后使用 HTMLEditorKit. Parser 对象的 parse( )方法。以下代码就是实现这些功能。

```
try{
URLConnection connection=
url.openConnection( ); //建立联接
  InputStream is=
Connection.getInputStream( );
    //实现输入流对象
Reader r=new InputStreamReader(is);
    //构建阅读器对象
HTMLEditorKit.Parser parse=new
HTMLParse( ).getParser( );
    Parse.parse(r,new Parser(url),true); //解析 HT-
ML
    }
    Catch (IOException e){
    Return;}
```

3.2 ParserCallback类定义及回调方法的实现

HTMLEditorKit. Parser 对象在解析 HTML 文件的时候，通常是按照 HTML 文档源文件中的字符顺序进行处理，遇到不同的字符将会调用相应的回调方法，在此我们考虑标签和文本的问题，其实 HTML 文件中的链接锅样是作为文本 TEXT 处理的。我们在回调方法 handleSimpleTag 方法中试图取得标签(文本除外)的链接 URL，所以处理过程中每次遇到开始标签和简单标签时，总是希望取得相应的 URL，而对于标签<h1>和
等，读到的字符串为 null，在此情况下使 handleSimpleTag 方法返回，否则将不会正常工作。类 handleStartTag 方法只是调用了 handleSimpleTag 方法。

再者，这里是简单的示例，要求将网页 HTML 文件中的超级链接名及其 URL 对应读出，我们在 ParserCallback 类中定义一个用于同步的布尔变量，只要 handleSimpleTag 方法处理到标签<a href……>就使 isLink 值为真 true。然后 HTMLEditorKit. Parser 对象处理到超级链接名文本时，调用回调方法 handleText 时，只有当 isLink 值为真 true 时的文本读出，处理完此文本后自动 isLink 复位 false 值，从而实现它们一一对应的关系。以下是我们用到的 ParserCallback 类定义(省去某些回调方法)：

```
Protected class Parser extends
HTMLEditorKit.ParserCallback {
    protected URL base;
    Boolean isLink=false;
    Public Parser(URL base)
    {
    This.base=base;} //ParserCallback 构造函数
    Public void handleText(char[ ]data,int pos)
    {
    String str2=String.valueOf(data);
    if (isLink) //将链接名写入容器
    workname.add(str2); }
    public void handleSimpleTag(HTML.Tag t,Muta-
bleAttributeSet a,int pos) {
    String href=
(String a.getAttribute(HTML.Attribute.HREF);
    if
    (href= =null||(href.indexOf (http: //)= = -1))
Return;
    Else isLink=true;
    Workurl.add(href); }
Public void handleStarTag(HTML. Tag t,MutableAt-
tributeSet a,int pos) {
    handleSimpleTag(t,a,pos); }}
```

4 结论

Swing 中包含了完整的 HTML 解析器功能，用户可以使用它进行开发，来解析 HTML 文档，从而达到灵活适用的目的。使用 HTMLEditorKit. Parser 编程在处理一些不甚规范的 HTML 文档尤其有益，有些现成的 HTML 解析器在解析不甚规范的 HTML 文档的技术的基础，笔者使用 Jbuilder8 就上述阐述到简单的功能进行了实现，供大家参考。当然解析 HTML 文件不是一件简单的事件，对复杂的网页 HTML 文件通过 Java 编程进行解析将有很多工作要做。

参考文献

1 美 Heaton，J.著，童兆凤等译，网络机器人 Java 编程指南，电子工业出版社，2002.7。

2 Elliotte Rusty 著，刘东华等译，Java 网络编程，中国电力出版社，2001。8。

3 美，Jacquie Barker 著，韩柯等译，Java 面向对象编程指南，电子工业出版社，2001。11。