COMP4321 Lab 2 - HTML Parser

1 Introduction

HTML Parser is a Java library used to parse HTML. It manages the connections and provides functions for parsing the html. It handles two use-cases: extraction and transformation. Extraction means extracting text, links, resources, etc., from a webpage, and transformation means transforming a webpage to another. To build a spider for a web search engine, we only need to use a small part of the library.

2 Packages

- org.htmlparser - The basic API classes which will be used by most developers when working with the HTML Parser.
- org.htmlparser.beans - The beans package contains Java Beans using the HTML Parser.
- org.htmlparser.filters - The filters package contains example filters to select only desired nodes.
- org.htmlparser.http - The http package is responsible for HTTP connections to servers.
- org.htmlparser.lexer - The lexer package is the base level I/O subsystem.
- org.htmlparser.lexerapplications.tabby - The Tabby program is a demonstration of how to use the underlying Lexer classes to perform file I/O.
- org.htmlparser.lexerapplications.thumbelina - Extract the images behind thumbnail images.
- org.htmlparser.nodes - The nodes package has the concrete node implementations.
- org.htmlparser.parserapplications - Example applications.
- org.htmlparser.parserapplications.filterbuilder
- org.htmlparser.parserapplications.filterbuilder.layouts
- org.htmlparser.parserapplications.filterbuilder.wrappers
- org.htmlparser.sax - The sax package implements a SAX (Simple API for XML) parser for HTML.
- org.htmlparser.scanners - The scanners package contains classes responsible for the tertiary identification of tags.
- org.htmlparser.tags - The tags package contains specific tags.
- org.htmlparser.util - Code which can be reused by many classes, is located in this package.
- org.htmlparser.util.sort - Provides generic sorting and searching.
- org.htmlparser.visitors - The visitors package contains classes that use the Visitor pattern.

3 Example

For any information about the library, visit the site http://htmlparser.sourceforge.net/index.html. Below are the sample programs that are useful for building a spider:

- StringExtractor.java
  Extract text from a web page.
- LinkExtractor.java
  Extract links/mail addresses from a web page.

To compile the examples, you need to download the package htmlparser1_6_20060610.zip and extract it. Then you can copy the jar files in htmlparser1_6/lib to a directory, say ” /lib”
The StringExtractor.java then can be compiled by ”javac -cp /lib/htmlparser.jar StringExtractor.java” and executed by ”java -cp /lib/htmlparser.jar:. StringExtractor”. (Replace : with ; for Windows)
You can find more examples in http://htmlparser.sourceforge.net/samples.html. The source codes for the examples can be found in src.zip in the package.
You will find that the links extracted by the provided LinkExtractor.java are relative. To get the absolute link, you can use the constructor of the URL.

```
URL absoluteLink = new URL(base,relativeLink);
```

where base is the URL of the current visited webpage.
Here is another example using org.htmlparser.beans.HTMLLinkBean to extract links: TestLinks.java

4 Exercise

Requirements In your program, you are expected to mainly implement two functions:

1. public Vector extractWords();// Extract the words in the webpage and put it into a Vector You can use org.htmlparser.beans.StringBean to get the texts in the webpage and then use java.util.StringTokenizer to tokenize the words for further indexing.

2. public Vector extractLinks();// Extract the links in the webpage and put it into a Vector You can use org.htmlparser.beans.LinkBean to get the links in the webpage.

Start your work on the skeleton program Crawler.java. The following output is expected:

```
Words in http://www.cs.ust.hk/~dlee/4321/ (size = 688) :
This
course
homepage
is
accessible
...
from
previous
projects
are
allowed.



Links in http://www.cs.ust.hk/~dlee/4321/:
http://www.cse.ust.hk/~dlee/4321/index.html
http://www.cse.ust.hk/~dlee
https://canvas.ust.hk/courses/29940/pages/course-work-and-grading-scheme?module_item_id=442459
https://course.cse.ust.hk/comp4321/labs/project.html
http://www.cs.ust.hk/~dlee/4321/Password_Only/4321-topics.htm
https://canvas.ust.hk/courses/29940/pages/course-work-and-grading-scheme?module_item_id=442459
http://www.cse.ust.hk/~dlee/4321/references.htm
```

Hints

- You may use StringBean and LinkBean to get the desired result.
- You may use String.split to break the texts to tokens.

Submission
no need to submit