CS1003 Week 8 Exercise 2: XML Parsing

As with all lab exercises, this exercise is not assessed. It is intended solely to help you understand the module material.

In this exercise you will experiment with parsing XML and extracting information from XML using a DOM based approach. We will use the web API provided by Wikipedia, which is documented here on the "MediaWiki API help" page: https://en.wikipedia.org/w/api.php

Wikipedia has a very large API, but do not be scared! We will only use a very small part of the API.

The "languagesearch" action

One of the actions provided by the Wikipedia API is language search. You may call this API action by using a URL of the following form:

https://en.wikipedia.org/w/api.php?format=xml&action=languagesearch&search=english

Here, the base of the URL is "https://en.wikipedia.org/w/api.php" and it takes 3 parameters: "format", "actions", and "search". The values for these parameters are "xml", "languagesearch", and "english" respectively.

Go to this URL in your web browser. You should see some XML; this is the response to your query. Try changing the "english" with the name of some other language and see how the response changes. Try a word that is not the name of a language, what happens now?

The "opensearch" action

Another action provided by the Wikipedia API is open search. You can find this action in the list of all actions on the main help page as well. The open search action allows us to search the whole of Wikipedia for a given query. Following is an example URL for searching for the keyword "linux":

https://en.wikipedia.org/w/api.php?format=xml&action=opensearch&search=linux

This URL is similar in structure to the URL for the "languagesearch" action.

Go to this URL in your browser. The XML response served by Wikipedia is a bit more complicated. There are more levels of nesting.

(More info about this action here: https://www.mediawiki.org/wiki/API:Opensearch)

Constructing a URL

To make an API call, you must first construct a URL object. A URL object can be constructed from a String:

```
URL url = new URL("....");
```

Parsing the XML response

We saw an example of parsing XML using a DOM based approach in the W03-2-XML example. Find this example on StudRes here: https://studres.cs.st-andrews.ac.uk/CS1003/Examples/W03-2-XML/XMLPrune.java

We will follow a similar approach here. Find the documentation for the parse method in the DocumentBuilder class in the Java API Specification. There are several overloaded versions of this method. The one used in the example from week 3 takes a File object as an argument. You can do the same for this exercise, but that would require downloading the XML response and saving it as a file. Alternatively, you can use another overloading og the parse method that takes an InputStream object as an argument.

Hint: The URL class has a method called openStream that you may find useful here.

Tasks

- Go through the "Reading XML Data into a DOM" section of the Java tutorial from the Oracle Java documentation.
  https://docs.oracle.com/javase/tutorial/jaxp/dom/readingXML.html
- Extract the "Description" fields from the results of an "opensearch" API call.
  Some methods you may want to use are the following.
  - getDocumentElement
  - getChildNodes
  - getNodeName
  - getNodeValue
  - getAttribute
- Create a program which takes a search query as a command line argument and displays the description fields from the response. Make sure your program handles search queries that are made of multiple words and/or contain non-ASCII characters.
  Hint: Use the URLEncoder.encode method when constructing the URL programmatically.
- If you have time, try some of the other actions from the MediaWiki API. Some of the actions require authentication (logging in), an interesting one that doesn't require authentication is "query".