Homework 3

January 23, 2017

The integrity of data is a cornerstone of all database usage. In this assignment, we will be converting data from CSV and XML formats into JSON. The point of this exercise is to demonstrate how to translate data from one format to another for loading into a database engine. In future assignments, we will use a real database and need to load data from various sources and formats.

1 CSV Data

In the assignment zip file, you will find “twitter.csv”. Write a CSV to JSON converter that outputs valid JSON. Use already built libraries to accomplish the task of reading CSV and writing JSON!

Your program should accept a CSV file name as an argument and output the valid JSON to standard output. An example is included as “formatdata”. You may run *fromatdata.exetwitter.csv* on a console and see the JSON printed in return. This program will detect what kind of file you are trying to format based on its extension.

2 XML Data

Just as with the CSV data, you will find “twitter.xml” in the zip. This file is correct. Write an XML to JSON converter that outputs valid JSON. Use already build libraries to accomplish the task of reading XML and writing JSON!

Your program should accept an XML file name as an argument and output the valid JSON to standard output. An example is included as “formatdata”. You may run *fromatdata.exe twitter.xml* on a console and see the JSON printed in return. This program will detect what kind of file you are trying to format based on its extension.

3 Validation of the data

Provided in the assignment zip file is a program called *testdata*. This program can be used to test the output of your converter. It will take valid JSON data on STDIN and output the number of valid tweets that were converted. If the JSON is invalid, it will print out an error indicating what might be wrong.

Your output data must match the format of the "sample.json" file! If any keys are differently capitalized, spelled, or omitted, the converter may not function properly. For missing values, assume 0 or 0.0 is valid for numbers and an empty string, "", is valid for missing strings.

There should be 150 tweets. when the formatter and validator are working correctly. Here is a sample:

```
./formatdata.osx twitter.csv | ./testdata.osx
2017/01/23 13:37:23 Successfully read 150 tweets.
```

4 Tips

Python 3 Resources

Refer to the CSV, XML, and JSON modules in the Python standard library. The Dive Into Python book has useful information on parsing XML in Python 3.

- http://www.diveintopython3.net/files.html
  - Basic file I/O with Python
- https://docs.python.org/3/library/csv.html
- https://docs.python.org/3/library/json.html
- http://www.diveintopython3.net/serializing.html#json-dump
  - This tutorial specifically shows you how to print JSON data from Python
- http://www.diveintopython3.net/xml.html#xml-parse
  - This tutorial specifically shows you how to read XML data in Python

Java Resources

- You will need to create a Java object to hold the data read in from various formats. The Jackson JSON library can write an array of objects that you create to write a valid JSON document.
- https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html
  - Maven is a useful tool for building Java projects with dependencies. A sample Maven project is included in the zip file with the Jackson JSON library and the opencsv CSV library included.
- https://www.tutorialspoint.com/java_xml/java_dom_parse_document.htm

  - XML parsing is built into the Java standard libary. This is a tutorial on its usage.

- http://tutorials.jenkov.com/java-json/jackson-installation.html
  - Using the sample project, you should already have Jackson, but here are installation instructions if not.

- http://tutorials.jenkov.com/java-json/jackson-objectmapper.html
  - This is a tutorial on using Jackson to output Java objects as JSON

- http://opencsv.sourceforge.net/
  - This is a project that allows you to read CSV data.

- https://sourceforge.net/p/opencsv/source/ci/master/tree/examples/AddressExample.java
  - An simple example of opencsv in use.

General Tips

- Create smaller sample files to work with. Two or three tweets probably scales up to 150 tweets and is easier to test. When your program works with the smaller sample, scale it up to all of the tweets. Just be sure your sample file reads valid with the "formatdata" program.

- Do not attempt to write your own XML/CSV/JSON parsers or writers. Use what is available as a library or built into the programming language.

- Use the example *formatdata* program to ensure that the CSV file is valid.

- Discuss any problems you run into on Piazza.

- Feel free to use any programming language you like to do this. C#, Ruby, Perl, etc. all have great libraries for reading these formats.

- Running the sample programs:
  - If you are on Windows, you should be able to run any of the .exe files in the Command Line (cmd.exe).
  - If you are on Linux, you should be able to run any of the .linux files in a terminal. You may have to run $chmod + x\ file.linux$.
  - If you are on OS X, you should be able to do the same as with the Linux binaries except use the .osx files.

5 What to turn in

Please turn the following into Canvas; you may zip the files up beforehand:

1. The source to your CSV to JSON converter

2. The source to your XML to JSON converter

3. A document describing how to build and run your converters

Be sure to turn in as much as you have at the end of the assignment. Partial credit is awarded based on completion, but late work is not acceptable.