XMLFlattener in Scala

Hesiquio ballines

Description

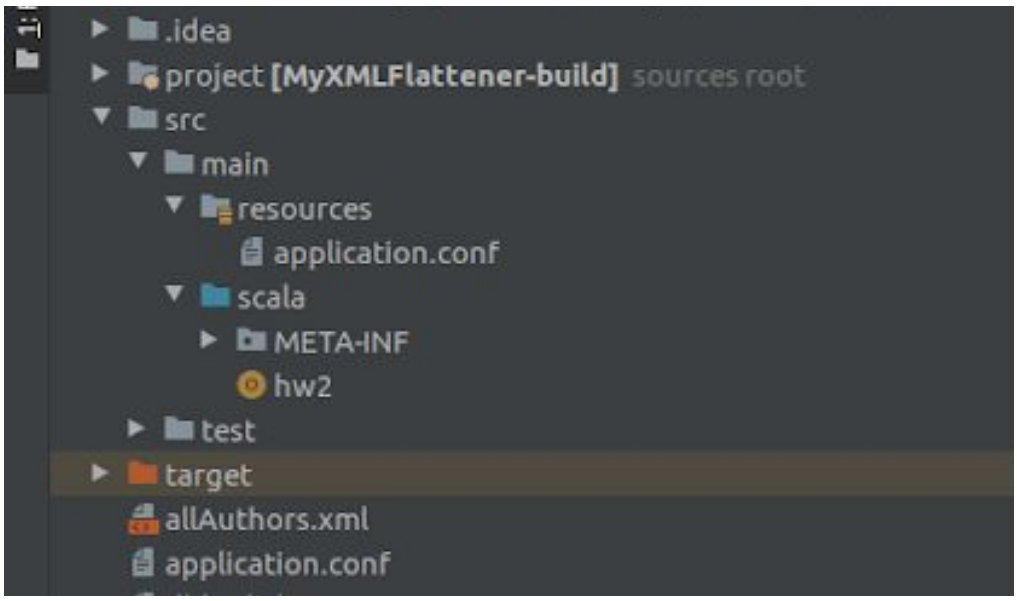
The XMLFlattener is designed to run on intellij. What it does is it uses end tags to put one tag and its contents into one line. The original intention is for use with hadoop map/reduce so that you can use the default input line format. For some it may be easier to process it one line at a time as there might be line by line readers that would like one entry per line.

With that being said if you are using Hadoop Map/reduce with XML I would recommend creating your own XMLInputFormat or using the Mahout implementation here: https://github.com/apache/mahout/blob/66f164057e322d2e63ea02c35c9e30c3969e80b1/integration/src/main/java/org/apache/mahout/text/wikipedia/XmlInputFormat.java

However, a limitation with Mahout's implementation is the lack of support for multiple tag parsing. For multiple tags I would recommend using Mohammed's Implementation here: https://github.com/Mohammed-siddiq/hadoop-XMLInputFormatWithMultipleTags

Finally I have included example from and to files so you can see what the program produces based on the input and config files.

Project Structure



XML Processing

The XMLFlattener works simply by looking for specific end tags that can be entered in the config file as "tags". In order for the flattener to work properly there can be a max of one end tag per line and the XML must be well formed. The best format would be one tag but one end tag per line should work.

Config Format

Both filename and output can be directories but they should be xml files, text files are accepted.

tags : “</endtag1>|</endtag2>|...</endtagN>”
filename: “input.xml”
outputfile: “output.xml”

XML Input Example

“Input.xml”, a bit larger than what is shown but this is an example taken from DBLP Public Dataset

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dblp SYSTEM "small.dtd">
<dblp>
<article mdate="2018-01-07" key="tr/meltdown/s18" publtype="informal">
        <author>Paul Kocher</author>
        <author>Daniel Genkin</author>
        <author>Daniel Gruss</author>
        <author>Werner Haas</author>
        <author>Mike Hamburg</author>
        <author>Moritz Lipp</author>
        <author>Stefan Mangard</author>
        <author>Thomas Prescher 0002</author>
        <author>Michael Schwarz 0001</author>
        <author>Yuval Yarom</author>
<title>Spectre Attacks: Exploiting Speculative Execution.</title>
<journal>meltdownattack.com</journal>
<year>2018</year>
<ee>https://spectreattack.com/spectre.pdf</ee>
</article>
<article mdate="2018-01-07" key="tr/meltdown/m18" publtype="informal">
        <author>Moritz Lipp</author>
        <author>Michael Schwarz 0001</author>
        <author>Daniel Gruss</author>
        <author>Thomas Prescher 0002</author>
        <author>Werner Haas</author>
        <author>Stefan Mangard</author>
        <author>Paul Kocher</author>
        <author>Daniel Genkin</author>
        <author>Yuval Yarom</author>
        <author>Mike Hamburg</author>
<title>Meltdown</title>
<journal>meltdownattack.com</journal>
<ee>https://meltdownattack.com/meltdown.pdf</ee>
<year>2018</year>
</article>
<book mdate="2019-05-27" key="tr/acm/CS2013">
<title>Computer Science Curricula 2013</title>
<publisher>ACM Press and IEEE Computer Society Press</publisher>
<year>2013</year>
<ee>https://doi.org/10.1145/2534860</ee>
<isbn>978-1-4503-2309-3</isbn>
</book>
<article mdate="2017-06-08" key="tr/gte/TR-0263-08-94-165" publtype="informal">
<author>Frank Manola</author>
<title>An Evaluation of Object-Oriented DBMS Developments: 1994 Edition.</title>
<journal>GTE Laboratories Incorporated</journal>
<volume>TR-0263-08-94-165</volume>
<month>August</month>
<year>1994</year>
<url>db/labs/gte/index.html#TR-0263-08-94-165</url>
</article><article mdate="2019-05-27" key="tr/gte/TR-0222-10-92-165" publtype="informal">
<author>Michael L. Brodie</author>
```

XML Output Example