

Apache CAMEL CSV to XML Converter

Muthuvel Periyasamy

Introduction	3
Project Solution Overview	3
How to Run the Project as Standalone Application	4

Introduction

Requirement is process CSV file which contains below structure

Germany,Berlin,81770900
Germany,Frankfurt,81770900
Germany,Munich,81770900

When you run the apache camel application it should convert it to below structure and place it into **Germany.xml** file as shown below.

```
<cities>
  <city name="Berlin">
    <population>81770900</population>
    <longitude>51.0</longitude>
    <latitude>9.0</latitude>
  </city>
  <city name="Frankfurt">
    <population>81770900</population>
  </city>
  <city name="Munich">
    <population>81770900</population>
  </city>
</cities>
```

Project Solution Overview

This is maven based java project , you can build it using any IDE or via command line.

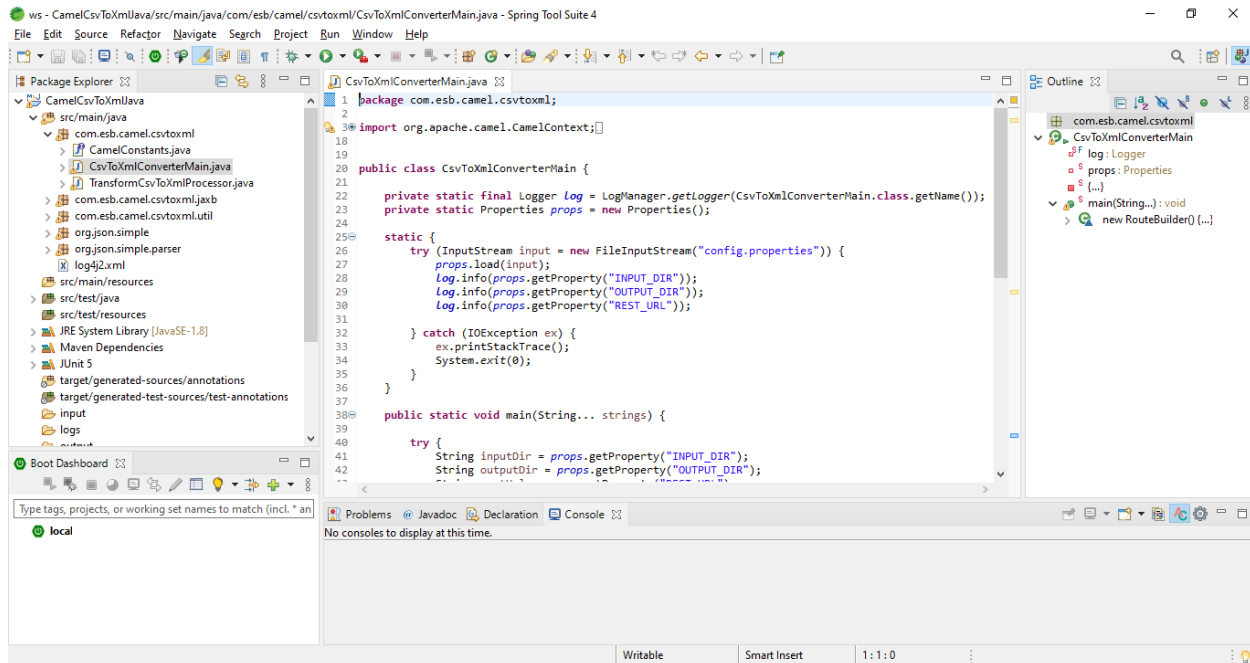
CsvToXmlConverterMain.java is the main java class , which starts reading the CSV file and configuration , and hands them over to the Processor class.

TransformCsvToXmlProcessor.java - gets the CSV inputs from above step.

- a) It load the country, capital ,population , longitude and latitude details from <https://restcountries.eu/rest/v2/all> only country capitals longitude and latitude, so in the

resulting xml , only country capitals will have longitude and latitude. Other cities will not have these details.

- b) Split the csv files to find the country , city and population
- c) Generate xml output with the help of JAXB classes.
- d) Writes the xml output in the name of <Country>.xml file in the output directory.
- e) Input CSV file will be deleted immediately after processing , for ease of testing.
- f) Output *.xml files will be generated inside the output directory with date time of the application running.



How to Run the Project as Standalone Application

- 1) Build the project using below command
maven clean install

You will get a success result as shown below.

```

Command Prompt
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/37/commons-parent-37.pom (0 B at 0 B/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-compat/3.3.9/maven-compat-3.3.9.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-compat/3.3.9/maven-compat-3.3.9.pom (3.8 kB at 13 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/3.0/maven-dependency-tree-3.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/3.0/maven-dependency-tree-3.0.pom (7.4 kB at 26 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/18.0/guava-18.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.25/slf4j-api-1.7.25.jar
Downloaded from central: https://repo.maven.apache.org/maven2/biz/aQute/bnd/biz.aQute.bndlib/5.1.1/biz.aQute.bndlib-5.1.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/osgi/org.osgi.core/6.0.0/org.osgi.core-6.0.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.25/slf4j-api-1.7.25.jar (41 kB at 61 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.4/commons-lang3-3.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/felix/maven-bundle-plugin/5.1.1/maven-bundle-plugin-5.1.1.jar (213 kB at 136 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-compat/3.3.9/maven-compat-3.3.9.jar
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/guava/guava/18.0/guava-18.0.jar (2.3 MB at 1.0 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/4.2.0/plexus-archiver-4.2.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-compat/3.3.9/maven-compat-3.3.9.jar (290 kB at 105 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/3.0/maven-dependency-tree-3.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-archiver/4.2.0/plexus-archiver-4.2.0.jar (196 kB at 64 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-30/plexus-container-default-1.0-alpha-30.jar
Downloaded from central: https://repo.maven.apache.org/maven2/biz/aQute/bnd/biz.aQute.bndlib/5.1.1/biz.aQute.bndlib-5.1.1.jar (3.7 MB at 1.2 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-dependency-tree/3.0/maven-dependency-tree-3.0.jar (37 kB at 12 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-lang3/3.4/commons-lang3-3.4.jar (435 kB at 129 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/osgi/org.osgi.core/6.0.0/org.osgi.core-6.0.0.jar (475 kB at 139 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-30/plexus-container-default-1.0-alpha-30.jar (237 kB at 68 kB/s)
[INFO] Building jar: D:\workspaces\ eclipse -ws\CamelCsvToXmlJava\target\CamelCsvToXmlJava-jar-with-dependencies.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ CamelCsvToXmlJava ---
[INFO] Installing D:\workspaces\ eclipse -ws\CamelCsvToXmlJava\target\CamelCsvToXmlJava.jar to C:\Users\Lenovo\.m2\repository\CamelCsvToXmlJava\CamelCsvToXmlJava\0.0.1-SNAPSHOT\CamelCsvToXmlJava-0.0.1-SNAPSHOT.jar
[INFO] Installing D:\workspaces\ eclipse -ws\CamelCsvToXmlJava\pom.xml to C:\Users\Lenovo\.m2\repository\CamelCsvToXmlJava\CamelCsvToXmlJava\0.0.1-SNAPSHOT\CamelCsvToXmlJava-0.0.1-SNAPSHOT.pom
[INFO] Installing D:\workspaces\ eclipse -ws\CamelCsvToXmlJava\target\CamelCsvToXmlJava-jar-with-dependencies.jar to C:\Users\Lenovo\.m2\repository\CamelCsvToXmlJava\CamelCsvToXmlJava\0.0.1-SNAPSHOT\CamelCsvToXmlJava-0.0.1-SNAPSHOT-jar-with-dependencies.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 42.129 s
[INFO] Finished at: 2021-06-10T23:54:44+04:00
[INFO]
D:\workspaces\ eclipse -ws\CamelCsvToXmlJava>

```

2) Copy the CamelCsvToXmlJava-jar-with-dependencies.jar generated in above steps to directory

Also copy **config.properties** , **log4j2.xml** , **data.csv** and **data2.csv** files

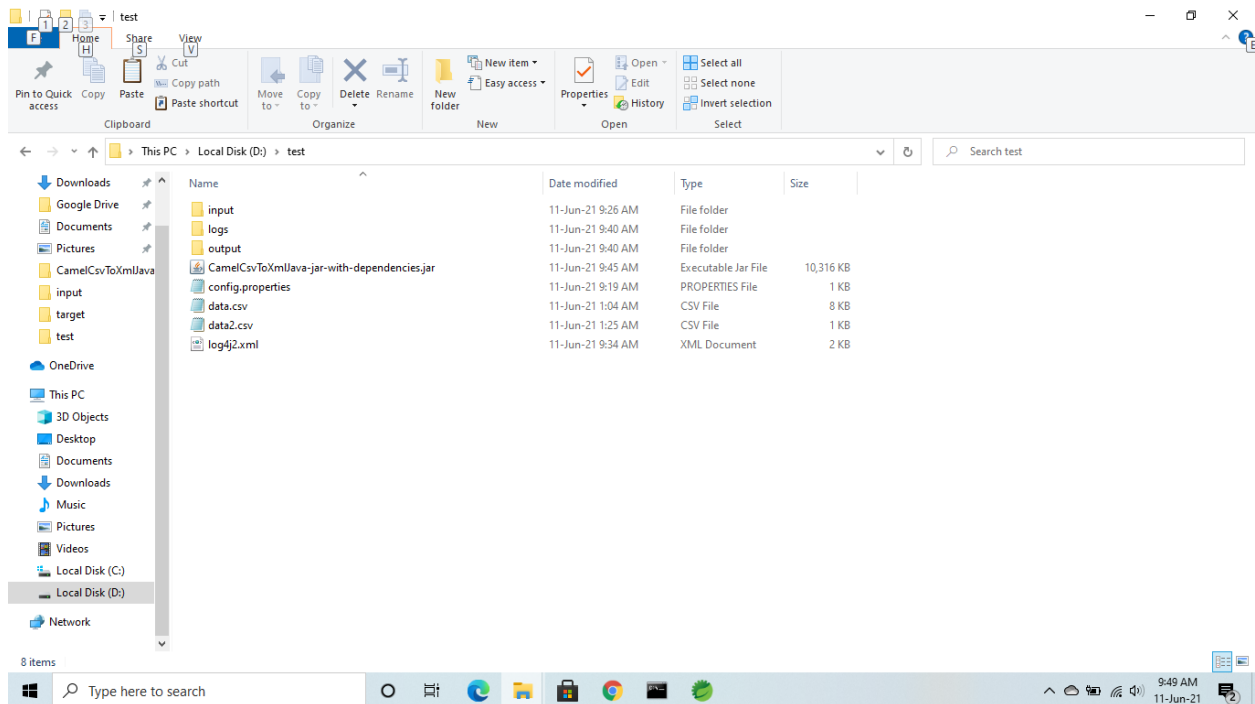
a) Config.properties - must have the details of input and output directories , along with Rest Url as shown below.

INPUT_DIR=input

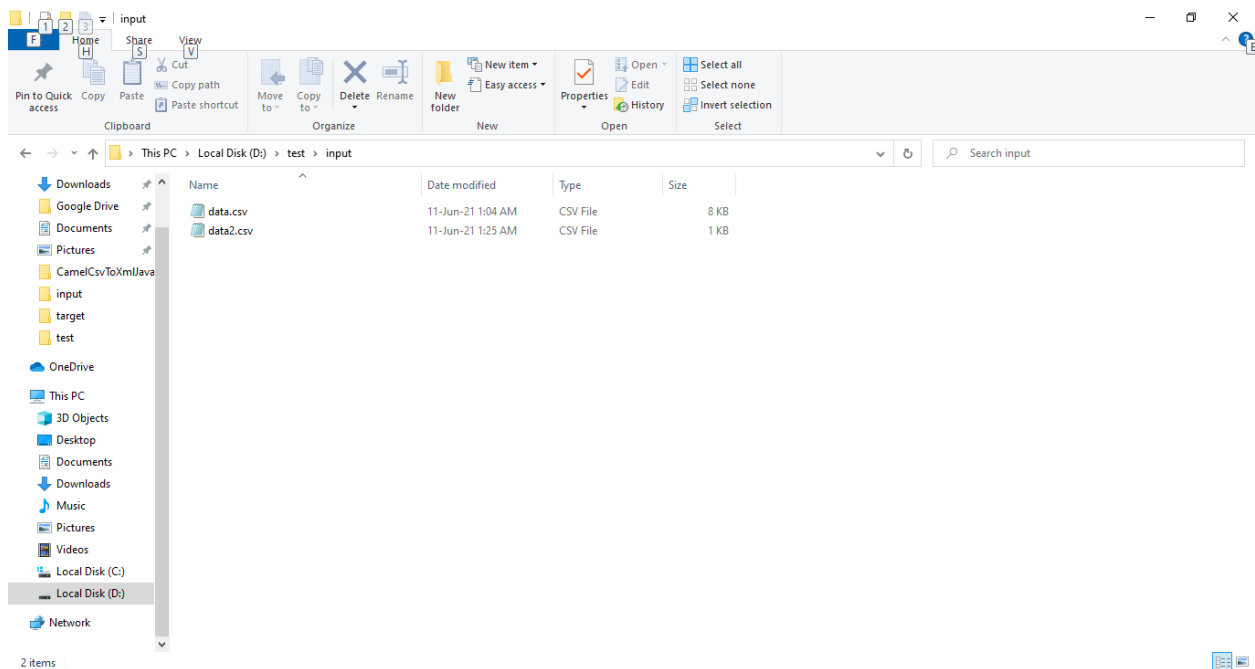
OUTPUT_DIR=output

REST_URL=https://restcountries.eu/rest/v2/all

b) Create input , output directories in the same location , as shown below.



c) Copy the csv file to the input directory



d) Run the program using below command

java -Dlog4j.configurationFile=log4j2.xml -jar CamelCsvToXmlJava-jar-with-dependencies.jar

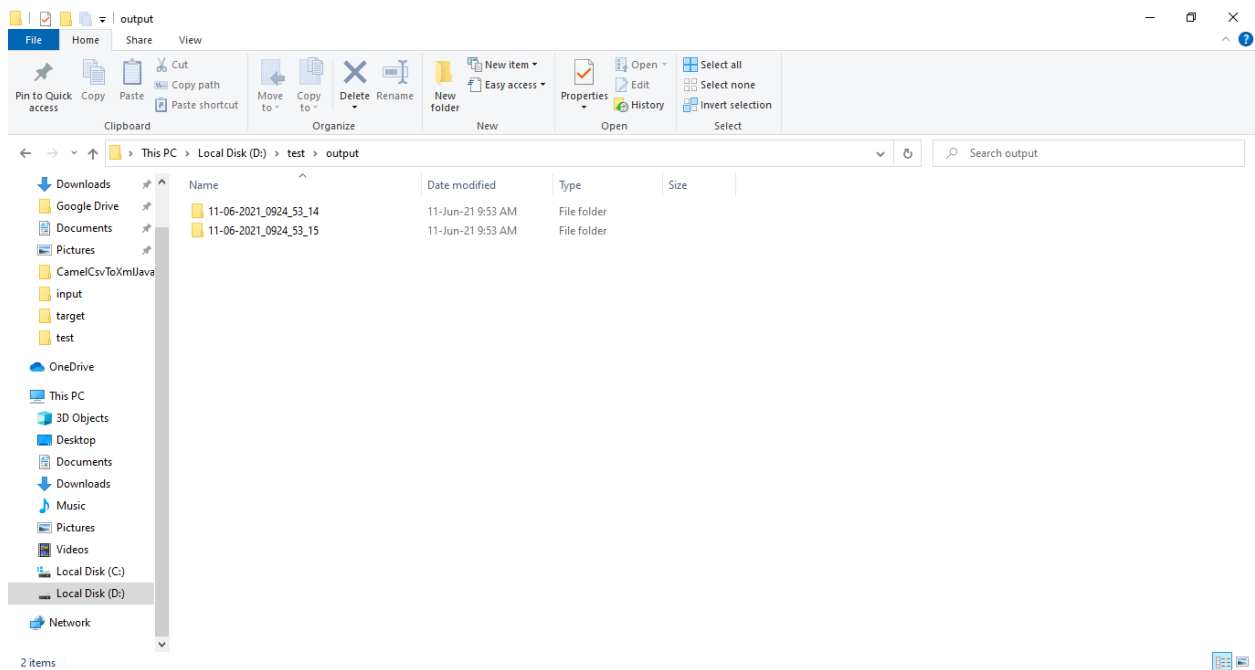
You will get output as shown below.

```

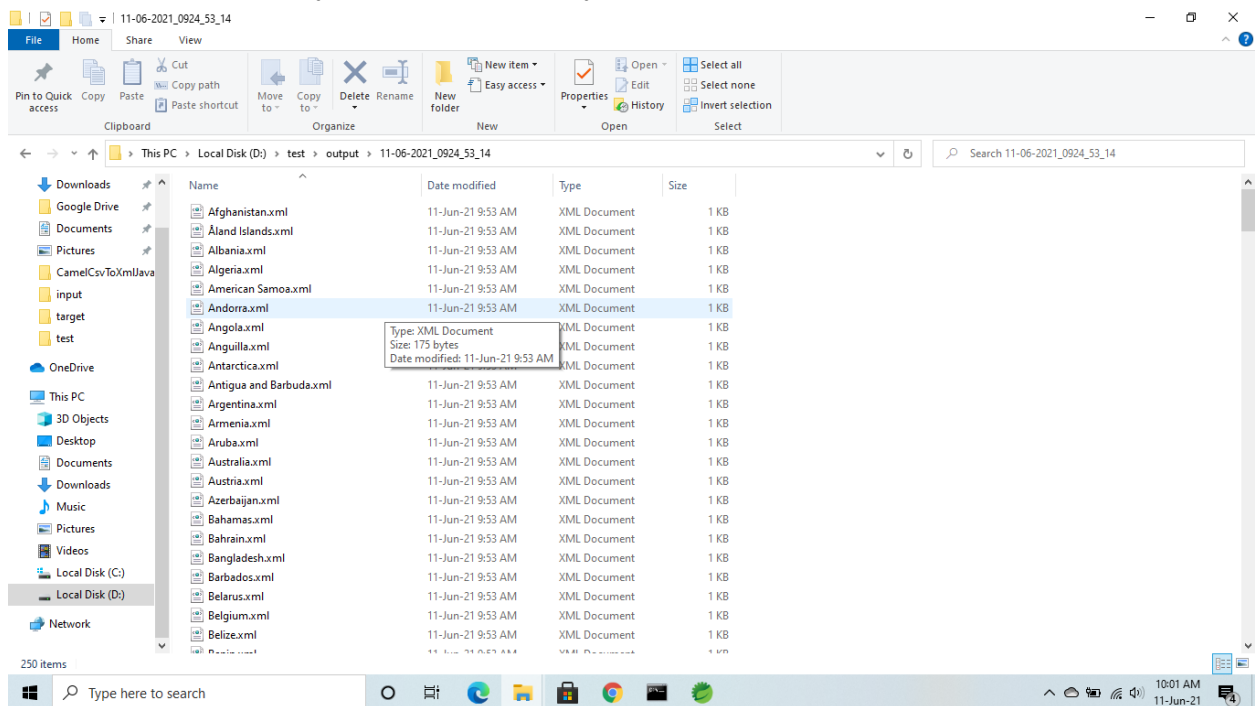
D:\test>java -Dlog4j.configurationFile=log4j2.xml -jar CamelCsvToXmlJava-jar-with-dependencies.jar
WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact performance.
2021-06-11 09:53:10,779 main DEBUG null null initializing configuration XmlConfiguration[location=D:\test\log4j2.xml]
2021-06-11 09:53:10,791 main DEBUG Installed 0 script engines
2021-06-11 09:53:10,792 main DEBUG PluginManager 'Core' found 122 plugins
2021-06-11 09:53:10,792 main DEBUG PluginManager 'Level' found 0 plugins
2021-06-11 09:53:10,797 main DEBUG PluginManager 'Lookup' found 15 plugins
2021-06-11 09:53:10,799 main DEBUG Building Plugin[name=layout, class=org.apache.logging.log4j.core.layout.PatternLayout
].
2021-06-11 09:53:10,812 main DEBUG PluginManager 'TypeConverter' found 26 plugins
2021-06-11 09:53:10,826 main DEBUG PatternLayout$Builder(pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n",
PatternSelector=null, Configuration(D:\test\log4j2.xml), Replace=null, charset="null", alwaysWriteExceptions="null", dis
ableAnsi="null", noConsoleNoAnsi="null", header="null", footer="null")
2021-06-11 09:53:10,827 main DEBUG PluginManager 'Converter' found 44 plugins
2021-06-11 09:53:10,828 main DEBUG Building Plugin[name=appender, class=org.apache.logging.log4j.core.appender.ConsoleAp
pender].
2021-06-11 09:53:10,837 main DEBUG ConsoleAppender$Builder(target="SYSTEM_OUT", follow="null", direct="null", bufferedIo
="null", bufferSize="null", immediateFlush="null", ignoreExceptions="null", PatternLayout(%d{HH:mm:ss.SSS} [%t] %-5level
%logger{36} - %msg%n), name="LogToConsole", Configuration(D:\test\log4j2.xml), Filter=null, ={})
2021-06-11 09:53:10,840 main DEBUG Starting OutputStreamManager SYSTEM_OUT.false.false
2021-06-11 09:53:10,840 main DEBUG Building Plugin[name=layout, class=org.apache.logging.log4j.core.layout.PatternLayout
].
2021-06-11 09:53:10,842 main DEBUG PatternLayout$Builder(pattern="%d %p %c{1.} [%t] %m%n", PatternSelector=null, Configu
ration(D:\test\log4j2.xml), Replace=null, charset="null", alwaysWriteExceptions="null", disableAnsi="null", noConsoleNoA
nsi="null", header="null", footer="null")
2021-06-11 09:53:10,852 main DEBUG Building Plugin[name=TimeBasedTriggeringPolicy, class=org.apache.logging.log4j.core.a
ppender.rolling.TimeBasedTriggeringPolicy].
2021-06-11 09:53:10,855 main DEBUG TimeBasedTriggeringPolicy$Builder(interval="null", modulate="null", maxRandomDelay="n
ull")
  
```

e) You can see the generate xmls files inside output folder in timestamp folder

As shown below



Inside timestamp folder your will find <country>.xml as show below



f) Open the xml files , you will find result as shown below.


```
<cities>
  <city name="Berlin">
    <population>81770900</population>
    <longitude>51.0</longitude>
    <latitude>9.0</latitude>
  </city>
  <city name="Frankfurt">
    <population>81770900</population>
  </city>
  <city name="Munich">
    <population>81770900</population>
  </city>
</cities>
```