PreProccessFilter 3

Introduction

This preprocess filter normalizes the syntax of an xml file by rearranging the structure of it in order to fix syntactic issues and to be compliant with a semantic model's concepts. Its main functionality is to split the values of an element, to created sub elements and assign these values to them. It is applied mostly in cases that the information about an event (actor, date, type) is mixed in an element and it needs to be spitted and assigned to the corresponding sub-elements.

The filter is implemented as a maven java project. It can be executed either by the command line, providing a number of arguments or by filling a configuration file and running the produced jar.

User Manual

The user can use the filter either by running it by the command line providing the required parameters or by filling a configuration file and running the jar. These approaches are described in the two following sections.

Running by the config.properties file and the jar

In order to use the filter by running the jar the user has to fill a config.properties file. Specifically the user has to fille the following:

- inputFilePath : the path of the xml input file
- outputFilePath: the path of the ouput xml file that will be produced
- parentNode: the parent node that contains the values that will be splitted and re-assigned
- newParentNode: the new parent node that will be created
- intermediateNode: the main intermediate node that will be the child of the new parent node
- intermediateNodes: the intermediate nodes (splitted by ",") that will be created and will be the children of the intermediate node and will contain the splitted values
- delimeter: the delimeter that will be used for splitting the initial values of the parent Node

After the filling of the config.properties file the user can run the jar (***preProcessFilter3-1.0.one-jar.jar***) either by double-clicking on it or by running it from the command file with the command:

java –jar preProcessFilter3-1.0.one-jar.jar

IMPORTANT: the config.properties file and the preProcessFilter3-1.0.one-jar.jar must be in the same folder

Running by using command line arguments

To run the filter by the command line the user has to provide the following arguments:

-inputFile : the path of the xml input file
-outputFile: the path of the ouput xml file that will be produced
-parentNode: the parent node that contains the values that will be splitted and re-assigned
-newParentNode: the new parent node that will be created
-intermediateNode: the main intermediate node that will be the child of the new parent node
-intermediateNodes: the intermediate nodes (splitted by ",") that will be created and will be the children of the intermediate node and will contain the splitted values
-delimeter: the delimeter that will be used for splitting the initial values of the parent Node

The command to run the filter is the following:

java -jar preProcessFilter3-1.0.one-jar.jar -inputFile "inputFilePath" -outputFile "outputFilePath" -parentNode "parentNodeName" -newParentNode "newParentNodeName" -intermediateNode "intemediateNode" -intermediateNodes "intNode1,intNode2,…" -delimeter "delimeter"

If the command runs successfully the following message will be returned:

"Successful Preprocessing!!!"

Example

An example of using the filter is described in this section.

The input file is the coins.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <Coin>
        <Acquisition>bequeathed; 1936-07-07; Young, Arthur W.</Acquisition>
        <Category>coin</Category>
        <Collection>Young Collection</Collection>
        <Date>48 B.C.</Date>
        <Maker>Caius Vibius Pansa Caetronianus;mint</Maker>
        <Material>silver</Material>
        <Name>Roman Republic; Seriesdenarius; denomination</Name>
    </Coin>
    <Coin>
        <Acquisition>bequeathed; 1936-07-07; Young, Arthur W.</Acquisition>
        <Category>coin</Category>
        <Collection>Young Collection</Collection>
        <Date>48 B.C.</Date>
        <Maker>Decimus Junius Brutus Albinus; moneyerRome; mintRome; mint</Maker>
        <Material>silver</Material>
```

```
        <Name>Roman Republic; Seriesdenarius; denomination</Name>
    </Coin>
</root>
```

The node of interest is highlighted with red color. This node refers to an acquisition event, but the date, the actor, and the type of the event are mixed. The goal is to split these values and create a new node called “AcquisitionEvents” that will have as children the nodes called again “Acquisition” that will have as children the new nodes “AcquisitionType”, ”Timespan”, ”Actor” that will contain the splitted values of the initial “Acquisition” node based on the “;” delimiter. The new xml file will be called output.xml

So, the configuration file will be the following:

```
#  Configuration file
#  If you want to use '\' make sure to change it '\\'
#  Separate the intermediateNodes by using ','

inputFilePath= coins.xml

outputFilePath= output.xml

parentNode=Acquisition

newParentNode=AcquisitionEvents

intermediateNode=Acquisition

intermediateNodes=AcquisitionType,Timespan,Actor

delimeter=;
```

By running the “***preProcessFilter3-1.0.one-jar.jar”*** the output.xml file will be produced (the new values are in green fonts):

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root>
      <Coin>
          <AcquisitionEvents>
            <Acquisition>
              <AcquisitionType>bequeathed</AcquisitionType>
              <Timespan>1936-07-07</Timespan>
              <Actor>Young, Arthur W.</Actor>
            </Acquisition>
          </AcquisitionEvents>
          <Category>coin</Category>
          <Collection>Young Collection</Collection>
          <Date>48 B.C.</Date>
          <Maker>Caius Vibius Pansa Caetronianus;mint</Maker>
          <Material>silver</Material>
          <Name>Roman Republic; Seriesdenarius; denomination</Name>
      </Coin>
```

```
    <Coin>
        <AcquisitionEvents>
          <Acquisition>
            <AcquisitionType>bequeathed</AcquisitionType>
            <Timespan>1936-07-07</Timespan>
            <Actor>Young, Arthur W.</Actor>
          </Acquisition>
        </AcquisitionEvents>
        <Category>coin</Category>
        <Collection>Young Collection</Collection>
        <Date>48 B.C.</Date>
        <Maker>Decimus Junius Brutus Albinus; moneyerRome; mintRome; mint</Maker>
        <Material>silver</Material>
        <Name>Roman Republic; Seriesdenarius; denomination</Name>
    </Coin>
</root>
```