A TEST ASSIGNMENT FOR THE POSITION JAVA / CLOJURE DEVELOPER

The input for the test tool is a logfile that can be downloaded from http://debreuck.neirynck.com/opdracht/server.zip. This is a plain text file (log file), where each line has a fixed format: <timestamp> [<thread>]<loglevel> [<message sender class>]: <log message>

The timestamp format is:

YYYY-MM-DD HH:mm:ss,SSS
(e.g. 2010-10-06 09:02:10,357)

The possible log levels are DEBUG, INFO, WARN or ERROR. The log message usually only covers a single line, but it is possible that multiple lines are used by a log message. For example, when printing an exception stack trace (see line 3627).

The logfile contains a snapshot of one of our server applications. The purpose of this application is to manage documents. These documents can be PDF files, in which case the client will ask the server to start a rendering. This is done through the command (aka. service) "startRendering", using the document id as argument (which is a number) and the page number (zero-based index). You can find this in the log file by searching for the string "Executing request startRendering". After this, the service will return a unique id (found by looking for "Service startRendering returned ..."). This service is executed in a single thread, so you can link the startRendering to its result by following the same thread. (Multiple requests may be executed at once but in different threads.) The client will keep executing, without waiting for the result of the rendering operation.
When the rendering has finished, the server will send an event to the client, mentioning the previously acquired unique id. The client knows that at this time, it can retrieve the result of the rendering operation. This is the service "getRendering" (using the unique id as argument). This service returns an InputStream with the rendered image as its contents. Such a rendering operation may take a long time (sometimes dozens of seconds). It is therefore possible that one client (or multiple clients) retrieve the same rendering (same document id, same page), before the rendering has completed. In this case the service will return the same unique id for this rendering. Then you should see multiple "getRendering" requests for this unique id.

The assignment is to create a tool that "scans" the logfile and extracts the following information from it:

- When was a startRendering command started, and for which document id?
- What was the unique id for this rendering command?
- Have there been any more startRendering commands in this period that have returned the same unique id?
- How many extra getRendering services have been executed for this UID and when was each of these services executed?

This information should be put into an XML file. The format of this file should look like this:

```
<report>
        <rendering>
                <!-- Document id -->
                <document>1234</document>
                <page>0</page>
                <!-- UID of the startRendering -->
                <uid>12345665456-4565</uid>
                <!-- One or more timestamps of the startRendering -->
                <start>2010-10-06 09:03:05,873</start>
                <start>2010-10-06 09:03:06,547</start>
                    ... (maybe some more starts)
                <!-- One or more timestamps of getRendering -->
                <get>2010-10-06 09:03:05,873</get>
                <get>2010-10-06 09:03:06,547</get>
                    ... (Possibly more gets)
        </rendering>
        <!-- Some more renderings... -->
          ...
        <!-- Summary -->
        <summary>
                <!-- Total number of renderings -->
                <count>40</count>
                <!-- Number of double renderings (multiple starts with same UID) -->
                <duplicates>1</duplicates>
                <!-- Number of startRenderings without get -->
                <unnecessary>0</unnecessary>
        </summary>
</report>
```

Please note that this logfile is a snapshot, so it's possible that at the start of the file there are some getRendering commands without an accompanying startRendering. For the same reason it's possible that at the end of the file there are startRendering commands without matching getRendering. The first scenario can be taken into account, the second cannot be predicted, so it may be ignored. In the report, this will be shown as a startRendering without getRendering.

It's also possible that there is not any startRendering that returns the same UID multiple times, of maybe that there are no starts that have nog associated gets. I don't know this logfile by heart ;-).

What I would like to receive is a small tool that takes as input this logfile, and generates as output an XML file that contains the above info. Please write it in either Java or Clojure (these are the main languages we use internally). You are also free to choose whether the app should be a command-line application or if it has a GUI frontend (but a web application seems overkill for this).
Of course I expect that you send me the source code and that I can compile and run it without much problems. It's up to you how you structure the code and where you give more or less attention to. What's important to me is the quality of the code, and not so much the "gold plating" of a fancy GUI :-).

It is not a very big application, so I think it must be possible to finish it in a week or two. The result can be send to wout@d-n.be. If there are any problems or questions, of course you may contact me.

Kind regards,
Wout Neirynck
CTO Debreuck Neirynck
wout@d-n.be