CUSTOM EVENT LOGGING FROM TEXT LOGFILE USING GRADLE

Massi Mapani - Software Engineer American Express UK

February 23, 2022

Abstract

This paper the solution for the Credit Suisse coding Assignment provided to me from Stanford Black on the 14th of February.

I. INTRODUCTION

A software framework gives a standardized method and structure for building, designing, testing and deploying code; a program or platform's architecture can be solely based on the choice of framework used [1]. Frameworks are reusable for functionality especially for large software platforms which are multi-layered and multi-platformed. Examples of software frameworks include compilers, code libraries and application programming interfaces (APIs). Frameworks have allowed developers to reuse implementations and architectures to selectively add functionality and customize features by inserting user written code [2].

This report provides details to the approach taken to solve the coding assignment.

II. BACKGROUND

Gradle is a build automation tool used for code build with an extensive amount of flexibility in-cooperated into it's design [3]. This coding challenge submission made use of Gradle for the build wrapper functionality and maven's mvn project management tool for the lifecycle management. In this report the method of implementing the logic will be explained and how it can be extended upon.

It is extremely essensial to get the project structure right to use these tools in a collaborative team format and thus a Github repository was made to reflect this.

III. SCOPING THE PROBLEM

The assignement required a solution that reads a text file filled with event logs. These event logs come as NoSQL JSON bodies with three mandatory attributes:

- The event id: which is assumed to be unique per executed event.
- The event state: this attribute can occur in a non logical flow per unique id key, hence an event id 'FINISHED' may be written into the logfile.txt before it's 'STARTED' counterpart.
- The event timestamp: The time when the event record was made for math calculation of the duration of the event

An event entry with the unique id attribute may or may not have 2 additional attributes which would assist with performance and functionality of the solution. These are:

- The event hostname: This attribute indicates which specific compute resource is running the event. This would be of interest for geolocation platforms, pricing and prioritizing purposes for high business value applications or compliance specific use cases
- The event log type: This is mainly used for Software as a Service architecture applications and being able to use this attributed for design and performance trade-offs is a great fintech approach provided that database schemas, data science principles and Machine learning are utilized hand in hand to create high performance platforms

IV. METHOD

To begin solving the problem; reading in the text file and converting each entry into JSON bodies from string data types is the first step. From [4] there are 3 approaches one can choose do this with; Gson from Google which is a high performance conversion function and is recommended if building an application in a decoupled micro service structure is the approach, JSON-Simple is another choice should memory constraints be a hard trade-off from a design perspective with client-server packets used for sending and receiving requests and lastly Jackson which is a oneline feature rich option that provides for lighting-fast real time processing depending on the application and business use case. This step requires intentional communication of business purpose functionality for an engineering decision to be made.

The next step is to calculate the duration of a complete event. This can be logically done with the pseudo code:

Algorithm 1 Duration

Input FilePath
OutputDuration, Alert
Text to JSON body
for $id, \ldots$ **do**
   if unique **then**
      populate statecounter array
      Begin a a separate thread PIDNo
Else
      Find thread PIDNo and put BooleanState
      for statecounter === 2 **do**
         PIDNo calculate
Duration = TIMESTAMPFINISHED - TIMESTAMPSTARTED
      if Duration≥ 4 **then**$Alert = True$
$RETURN(Duration, Alert)$
Else
$Alert = False$
$RETURN(Duration, Alert)$

The following step requires integration of Data Science principles such as Normalising the Data and creating creating STAR Schema's and really cool Data Science implementation given that HSQLDB has embedded testing and most importantly logging facility through the use of the library java.util.logging, auto-updated TIMESTAMP columns on row updates fir the unique id and API specifications for Log4J. However, given that data security is a consistent risk factor in Financial Services following the principles of Information Security and process of reporting [5]. HSQLDB has in built POST alerting notifications and exploring more into this would have been done had more time been spent on this problem.

It is worth noting that given automation and multi-threading can be used for this problem solution, making use of Collections Library for safe multi-threading would have been utilised per event state pair with mutual exclusion principles applied to avoid race-conditions and minimize synchronization cost. This article [6] explains in principle how to implement and the

reasons why safe-multithreading is essential. Routine testing can be used for the Alert functionality withing HSQLDB, given the return condition for the duration of the event could be true or false

V. Conclusion

This report presents a method and approach to this real world problem in financial service computing and architectural design. Further information for business purpose for the solution and elaboration on use cases and user journeys would provided more finite solution routes for the implementation, enhancement and maintenance of a custom platform that makes use of event logs.

REFERENCES

[1] V. Myllärniemi, S. Kujala, M. Raatikainen and P. Sevońn, "Development as a journey: factors supporting the adoption and use of software frameworks," Journal of Software Engineering Research and Development, vol. 6, p. 6, 2018.

[2] H Vocke, "The practical test pyramid" 26 February 2018. [Online]. Available: https://martinfowler.com/articles/practical-test-pyramid.html. [Accessed 23 February 2022].

[3] Gradle Inc, "What is Gradle" 2021. [Online]. Available: https://docs.gradle.org/current/userguide/what$_i s_g radle.html.[Accessed 23 February 2022]$.

[4] J Paul, "3 ways to convert String to JSON object in Java? Examples" 23 February 2022. Available: https://www.java67.com/2016/10/3-ways-to-convert-string-to-json-object-in-java.html.

[5] National Cyber Security Center, "Log4j vulnerability - what everyone needs to know", 23 February 2022. [Online]. Available: https://www.ncsc.gov.uk/information/log4j-vulnerability-what-everyone-needs-to-know. [Accessed 23 February 2022].

[6] Y Pirildak, "How to Use Java Collections Safely in Multi-Threaded Environments," , Level Up Coding Publication, 27 March 2020, Available: https://levelup.gitconnected.com/how-to-use-java-collections-safely-in-multi-threaded-environments-83c94be57ab8.