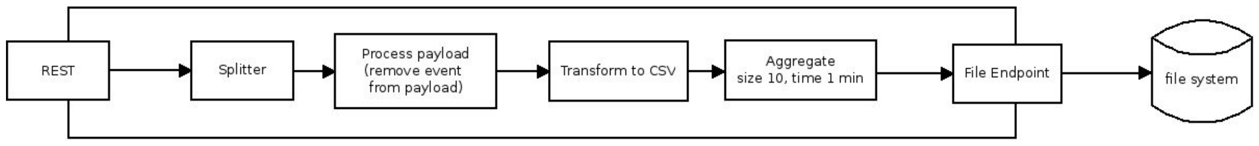
Problem Description
The system has a REST interface to accept a json payload (see attached sample payload). Each payload may contain one or more record. Each record needs to be extracted, remove the events portion of it and then transform it into CSV format. For performance reasons we want to batch (batch size 10) and then write to a file.

The above is a simplified real world use case that you will encounter in your day to day job. I have modeled the above problem as a Camel route. You need to learn and understand the basic camel concepts and use the Camel components to implement a camel route to do the above.

Challenge
Design and build a camel route to implement the above scenario as described below.


REST
Splitter
Process payload (remove event from payload)
Transform to CSV
Aggregate size 10, time 1 min
File Endpoint
file system

- All the camel concepts required can be learnt online.
- The camel route should run on SpringBoot. There are several tutorials online about how to use Camel with SpringBoot.
- You need to use Maven to compile and build the project.
- Logging information about the process flow is important as it helps trace issues in production. Use slf4j as the logging framework and log steps that you feel is important. The batch id should be used in all log messages. (hint keep the batchId as an exchange property so you can use it later)
- The route has a REST interface at the beginning of the route. The REST interface should be built with the Camel REST DSL.
- Each payload contains one or more records.
- We are using the splitter aggregator pattern (it’s common camel pattern).
- The splitter should split the records
- For each record get rid of the events section.
- Transform each record into CSV format.
- The aggregator should aggregate every 10 records or use a completion interval of 1 min (whichever happens first). Depending on the size of a payload, sometimes a single payload can be split into multiple aggregations or multiple payloads could be aggregated into one.
- Write each aggregation batch into a separate file.
- Write unit tests to test all the important pieces of the route. Look at wearBy, interceptors and mockEndpoints as possible strategies. Your test should use the CamelTestSupport class as the base class.
- Check in all code to a git hub repo.
- Add comments in your code where appropriate.