MODULE *ReportProcessorCommittingAtTheEnd*

EXTENDS *Integers*, *Sequences*, *TLC*, *Helpers*

VARIABLE *payReportQueue*, *processed*, *memoryDb*, *retryQueue*, *action*

$PayReportItems \triangleq \langle 1, 2, 3 \rangle$

$TypeOK \triangleq$
  $action \in \{$"consuming_pay_report", "processing_successfull", "put_in_memory_db", "put_in_retry_queue"$\}$

$Init \triangleq payReportQueue = PayReportItems$
  $\land \; processed = \langle\rangle$
  $\land \; memoryDb = \langle\rangle$
  $\land \; retryQueue = \langle\rangle$
  $\land \; action =$ "consuming_pay_report"

Function to add the item to the processed list, unless it's already there, simulating idempotency
$ProcessItem(item) \triangleq$ IF $item \notin SeqToSet(processed)$
  THEN $Append(processed, item)$
  ELSE $processed$

When a kafka message is processed successfully, we mark it as successfull, unless there is already a previous message in memory db, in this case we never process it, it will go to memory db as well directly
$ConsumePayReportSucc \triangleq payReportQueue \neq \langle\rangle$
  $\land \; memoryDb = \langle\rangle$
  $\land \; action =$ "consuming_pay_report"
  $\land \; action' =$ "processing_successfull"
  $\land \; processed' = ProcessItem(Head(payReportQueue))$
  $\land$ UNCHANGED $\langle payReportQueue, memoryDb, retryQueue \rangle$

When a kafka message is processed with error, we put the message in retry queue
$ConsumePayReportError \triangleq payReportQueue \neq \langle\rangle$
  $\land \; Head(payReportQueue) \notin SeqToSet(memoryDb)$
  $\land \; action =$ "consuming_pay_report"
  $\land \; action' =$ "put_in_memory_db"
  $\land \; memoryDb' = Append(memoryDb, Head(payReportQueue))$
  $\land$ UNCHANGED $\langle processed, payReportQueue, retryQueue \rangle$

After putting the message in memory db, we put it in retry queue
$PutInRetryQueue \triangleq action =$ "put_in_memory_db"
  $\land \; action' =$ "put_in_retry_queue"
  $\land \; retryQueue' = Append(retryQueue, Head(payReportQueue))$
  $\land$ UNCHANGED $\langle processed, memoryDb, payReportQueue \rangle$

We commit to kafka either after message is processed successfully OR after is had been put on the retry queue

$$
\begin{aligned}
CommitToPayReport \triangleq\ & (action = \text{“processing\_successfull”} \lor action = \text{“put\_in\_retry\_queue”}) \\
\land\ & action' = \text{“consuming\_pay\_report”} \\
\land\ & payReportQueue' = Tail(payReportQueue) \\
\land\ & \text{UNCHANGED } \langle processed,\ memoryDb,\ retryQueue \rangle
\end{aligned}
$$

We consume messaged from retry queue in parallel, it is only processed if sucessfully if the current item in the queue is the same as the *Head* of memory db, if so, the item is taken out of the retry queue and memory db, and added to processed list, if not, *ConsumeRetryQueueError* will be triggered and the item will end up in the end of the queue again

$$
\begin{aligned}
ConsumeRetryQueueSucc \triangleq\ & retryQueue \neq \langle\rangle \\
\land\ & Head(retryQueue) = Head(memoryDb) \\
\land\ & processed' = ProcessItem(Head(retryQueue)) \\
\land\ & memoryDb' = Tail(memoryDb) \\
\land\ & retryQueue' = Tail(retryQueue) \\
\land\ & \text{UNCHANGED } \langle action,\ payReportQueue \rangle
\end{aligned}
$$

We consume messaged from retry queue in parallel, if there is an error, the message goes back to the end of the retry queue

$$
\begin{aligned}
ConsumeRetryQueueError \triangleq\ & retryQueue \neq \langle\rangle \\
\land\ & retryQueue' = Append(Tail(retryQueue),\ Head(retryQueue)) \\
\land\ & \text{UNCHANGED } \langle processed,\ action,\ memoryDb,\ payReportQueue \rangle
\end{aligned}
$$

At any time, the consumer may panic in the middle and restart the process from the beginning

$$
\begin{aligned}
PanicConsumer \triangleq\ & action' = \text{“consuming\_pay\_report”} \\
\land\ & \text{UNCHANGED } \langle processed,\ payReportQueue,\ memoryDb,\ retryQueue \rangle
\end{aligned}
$$

$$
\begin{aligned}
Next \triangleq\ & ConsumePayReportSucc \\
\lor\ & ConsumePayReportError \\
\lor\ & PutInRetryQueue \\
\lor\ & ConsumeRetryQueueSucc \\
\lor\ & ConsumeRetryQueueError \\
\lor\ & CommitToPayReport \\
\lor\ & PanicConsumer
\end{aligned}
$$

$$
\begin{aligned}
& ProcessedCompleteAndInOrderInvariant \triangleq \\
& \quad \text{IF } Len(processed) > 0 \text{ THEN} \\
& \qquad \text{sequences are 1-indexed} \\
& \qquad \text{LET } F[i \in 1\ ..\ (Len(processed) + 1)] \triangleq \\
& \qquad\quad \text{IF } i > 1 \land \neg F[i-1] \text{ THEN FALSE} \\
& \qquad\qquad\qquad\quad \text{ELSE } processed[i] = PayReportItems[i] \\
& \qquad \text{IN } F[Len(processed)] \\
& \quad \text{ELSE TRUE}
\end{aligned}
$$

* Modification History
* Last modified *Wed Feb* 09 11:56:08 *CET* 2022 by *rchavesferna*
* Created *Thu Feb* 03 19:11:04 *CET* 2022 by *rchavesferna*