package com.twitter.unified\_user\_actions.enricher.driver

import com.twitter.unified\_user\_actions.enricher.internal.thriftscala.EnrichmentPlan

import com.twitter.unified\_user\_actions.enricher.internal.thriftscala.EnrichmentStage

import com.twitter.unified\_user\_actions.enricher.internal.thriftscala.EnrichmentStageStatus.Completion

import com.twitter.unified\_user\_actions.enricher.internal.thriftscala.EnrichmentStageStatus.Initialized

object EnrichmentPlanUtils {

implicit class EnrichmentPlanStatus(plan: EnrichmentPlan) {

/\*\*

\* Check each stage of the plan to know if we are done

\*/

def isEnrichmentComplete: Boolean = {

plan.stages.forall(stage => stage.status == Completion)

}

/\*\*

\* Get the next stage in the enrichment process. Note, if there is none this will throw

\* an exception.

\*/

def getCurrentStage: EnrichmentStage = {

val next = plan.stages.find(stage => stage.status == Initialized)

next match {

case Some(stage) => stage

case None => throw new IllegalStateException("check for plan completion first")

}

}

def getLastCompletedStage: EnrichmentStage = {

val completed = plan.stages.reverse.find(stage => stage.status == Completion)

completed match {

case Some(stage) => stage

case None => throw new IllegalStateException("check for plan completion first")

}

}

/\*\*

\* Copy the current plan with the requested stage marked as complete

\*/

def markStageCompletedWithOutputTopic(

stage: EnrichmentStage,

outputTopic: String

): EnrichmentPlan = {

plan.copy(

stages = plan.stages.map(s =>

if (s == stage) s.copy(status = Completion, outputTopic = Some(outputTopic)) else s)

)

}

def markStageCompleted(

stage: EnrichmentStage

): EnrichmentPlan = {

plan.copy(

stages = plan.stages.map(s => if (s == stage) s.copy(status = Completion) else s)

)

}

/\*\*

\* Copy the current plan with the last stage marked as necessary

\*/

def markLastStageCompletedWithOutputTopic(

outputTopic: String

): EnrichmentPlan = {

val last = plan.stages.last

plan.copy(

stages = plan.stages.map(s =>

if (s == last) s.copy(status = Completion, outputTopic = Some(outputTopic)) else s)

)

}

}

}