package com.twitter.product\_mixer.core.pipeline.pipeline\_failure

import com.fasterxml.jackson.core.JsonGenerator

import com.fasterxml.jackson.databind.JsonSerializer

import com.fasterxml.jackson.databind.SerializerProvider

import com.twitter.product\_mixer.core.model.common.identifier.ComponentIdentifierStack

private[pipeline\_failure] class PipelineFailureSerializer()

extends JsonSerializer[PipelineFailure] {

private sealed trait BaseSerializableException

private case class SerializableException(

`class`: String,

message: String,

stackTrace: Seq[String],

cause: Option[BaseSerializableException])

extends BaseSerializableException

private case class SerializablePipelineFailure(

category: String,

reason: String,

underlying: Option[BaseSerializableException],

componentStack: Option[ComponentIdentifierStack],

stackTrace: Seq[String])

extends BaseSerializableException

private def serializeStackTrace(stackTrace: Array[StackTraceElement]): Seq[String] =

stackTrace.map(stackTraceElement => "at " + stackTraceElement.toString)

private def mkSerializableException(

t: Throwable,

recursionDepth: Int = 0

): Option[BaseSerializableException] = {

t match {

case \_ if recursionDepth > 4 =>

// in the unfortunate case of a super deep chain of exceptions, stop if we get too deep

None

case pipelineFailure: PipelineFailure =>

Some(

SerializablePipelineFailure(

category =

pipelineFailure.category.categoryName + "/" + pipelineFailure.category.failureName,

reason = pipelineFailure.reason,

underlying =

pipelineFailure.underlying.flatMap(mkSerializableException(\_, recursionDepth + 1)),

componentStack = pipelineFailure.componentStack,

stackTrace = serializeStackTrace(pipelineFailure.getStackTrace)

))

case t =>

Some(

SerializableException(

`class` = t.getClass.getName,

message = t.getMessage,

stackTrace = serializeStackTrace(t.getStackTrace),

cause = Option(t.getCause).flatMap(mkSerializableException(\_, recursionDepth + 1))

)

)

}

}

override def serialize(

pipelineFailure: PipelineFailure,

gen: JsonGenerator,

serializers: SerializerProvider

): Unit = serializers.defaultSerializeValue(mkSerializableException(pipelineFailure), gen)

}