package com.twitter.product\_mixer.core.quality\_factor

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

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.MisconfiguredQualityFactor

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.PipelineFailure

case class QualityFactorStatus(

qualityFactorByPipeline: Map[ComponentIdentifier, QualityFactor[\_]]) {

/\*\*

\* returns a new [[QualityFactorStatus]] with all the elements of current QualityFactorStatus and `other`.

\* If a [[ComponentIdentifier]] exists in both maps, the Value from `other` takes precedence

\*/

def ++(other: QualityFactorStatus): QualityFactorStatus = {

if (other.qualityFactorByPipeline.isEmpty) {

this

} else if (qualityFactorByPipeline.isEmpty) {

other

} else {

QualityFactorStatus(qualityFactorByPipeline ++ other.qualityFactorByPipeline)

}

}

}

object QualityFactorStatus {

def build[Identifier <: ComponentIdentifier](

qualityFactorConfigs: Map[Identifier, QualityFactorConfig]

): QualityFactorStatus = {

QualityFactorStatus(

qualityFactorConfigs.map {

case (key, config: LinearLatencyQualityFactorConfig) =>

key -> LinearLatencyQualityFactor(config)

case (key, config: QueriesPerSecondBasedQualityFactorConfig) =>

key -> QueriesPerSecondBasedQualityFactor(config)

}

)

}

val empty: QualityFactorStatus = QualityFactorStatus(Map.empty)

}

trait HasQualityFactorStatus {

def qualityFactorStatus: Option[QualityFactorStatus] = None

def withQualityFactorStatus(qualityFactorStatus: QualityFactorStatus): PipelineQuery

def getQualityFactorCurrentValue(

identifier: ComponentIdentifier

): Double = getQualityFactor(identifier).currentValue

def getQualityFactor(

identifier: ComponentIdentifier

): QualityFactor[\_] = qualityFactorStatus

.flatMap(\_.qualityFactorByPipeline.get(identifier))

.getOrElse {

throw PipelineFailure(

MisconfiguredQualityFactor,

s"Quality factor not configured for $identifier")

}

}