package com.twitter.product\_mixer.core.service.async\_feature\_map\_executor

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.feature.featuremap.asyncfeaturemap.AsyncFeatureMap

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

import com.twitter.product\_mixer.core.service.Executor

import com.twitter.product\_mixer.core.service.Executor.\_

import com.twitter.product\_mixer.core.service.ExecutorResult

import com.twitter.stitch.Arrow

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class AsyncFeatureMapExecutor @Inject() (

override val statsReceiver: StatsReceiver)

extends Executor {

/\*\*

\* Forces an [[AsyncFeatureMap]] to hydrate and resolve into a [[FeatureMap]]

\* containing all [[com.twitter.product\_mixer.core.feature.Feature]]s that are

\* supposed to be hydrated before `stepToHydrateBefore`.

\*/

def arrow(

stepToHydrateFor: PipelineStepIdentifier,

currentStep: PipelineStepIdentifier,

context: Context

): Arrow[AsyncFeatureMap, AsyncFeatureMapExecutorResults] = {

Arrow

.map[AsyncFeatureMap, Option[Stitch[FeatureMap]]](\_.hydrate(stepToHydrateFor))

.andThen(

Arrow.choose(

Arrow.Choice.ifDefinedAt(

{ case Some(stitchOfFeatureMap) => stitchOfFeatureMap },

// only stat if there's something to hydrate

wrapComponentWithExecutorBookkeeping(context, currentStep)(

Arrow

.flatMap[Stitch[FeatureMap], FeatureMap](identity)

.map(featureMap =>

AsyncFeatureMapExecutorResults(Map(stepToHydrateFor -> featureMap)))

)

),

Arrow.Choice.otherwise(Arrow.value(AsyncFeatureMapExecutorResults(Map.empty)))

)

)

}

}

case class AsyncFeatureMapExecutorResults(

featureMapsByStep: Map[PipelineStepIdentifier, FeatureMap])

extends ExecutorResult {

def ++(

asyncFeatureMapExecutorResults: AsyncFeatureMapExecutorResults

): AsyncFeatureMapExecutorResults =

AsyncFeatureMapExecutorResults(

featureMapsByStep ++ asyncFeatureMapExecutorResults.featureMapsByStep)

}