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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.product\_mixer.core.model.marshalling.HasMarshalling

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

import com.twitter.product\_mixer.core.pipeline.candidate.CandidatePipelineBuilderFactory

import com.twitter.product\_mixer.core.service.candidate\_pipeline\_executor.CandidatePipelineExecutor

import com.twitter.product\_mixer.core.service.domain\_marshaller\_executor.DomainMarshallerExecutor

import com.twitter.product\_mixer.core.service.gate\_executor.GateExecutor

import com.twitter.product\_mixer.core.service.pipeline\_result\_side\_effect\_executor.PipelineResultSideEffectExecutor

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

import com.twitter.product\_mixer.core.service.query\_feature\_hydrator\_executor.QueryFeatureHydratorExecutor

import com.twitter.product\_mixer.core.service.selector\_executor.SelectorExecutor

import com.twitter.product\_mixer.core.service.transport\_marshaller\_executor.TransportMarshallerExecutor

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class MixerPipelineBuilderFactory @Inject() (

candidatePipelineExecutor: CandidatePipelineExecutor,

gateExecutor: GateExecutor,

selectorExecutor: SelectorExecutor,

queryFeatureHydratorExecutor: QueryFeatureHydratorExecutor,

asyncFeatureMapExecutor: AsyncFeatureMapExecutor,

domainMarshallerExecutor: DomainMarshallerExecutor,

transportMarshallerExecutor: TransportMarshallerExecutor,

pipelineResultSideEffectExecutor: PipelineResultSideEffectExecutor,

candidatePipelineBuilderFactory: CandidatePipelineBuilderFactory,

statsReceiver: StatsReceiver) {

def get[

Query <: PipelineQuery,

DomainResultType <: HasMarshalling,

Result

]: MixerPipelineBuilder[Query, DomainResultType, Result] = {

new MixerPipelineBuilder[Query, DomainResultType, Result](

candidatePipelineExecutor,

gateExecutor,

selectorExecutor,

queryFeatureHydratorExecutor,

asyncFeatureMapExecutor,

domainMarshallerExecutor,

transportMarshallerExecutor,

pipelineResultSideEffectExecutor,

candidatePipelineBuilderFactory,

statsReceiver

)

}

}