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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.product\_mixer.core.model.common.UniversalNoun

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

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

import com.twitter.product\_mixer.core.service.candidate\_decorator\_executor.CandidateDecoratorExecutor

import com.twitter.product\_mixer.core.service.candidate\_feature\_hydrator\_executor.CandidateFeatureHydratorExecutor

import com.twitter.product\_mixer.core.service.candidate\_source\_executor.CandidateSourceExecutor

import com.twitter.product\_mixer.core.service.filter\_executor.FilterExecutor

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

import com.twitter.product\_mixer.core.service.group\_results\_executor.GroupResultsExecutor

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class CandidatePipelineBuilderFactory @Inject() (

queryFeatureHydratorExecutor: QueryFeatureHydratorExecutor,

asyncFeatureMapExecutor: AsyncFeatureMapExecutor,

candidateDecoratorExecutor: CandidateDecoratorExecutor,

candidateFeatureHydratorExecutor: CandidateFeatureHydratorExecutor,

candidateSourceExecutor: CandidateSourceExecutor,

groupResultsExecutor: GroupResultsExecutor,

filterExecutor: FilterExecutor,

gateExecutor: GateExecutor,

statsReceiver: StatsReceiver) {

def get[

Query <: PipelineQuery,

CandidateSourceQuery,

CandidateSourceResult,

Result <: UniversalNoun[Any]

]: CandidatePipelineBuilder[

Query,

CandidateSourceQuery,

CandidateSourceResult,

Result

] = {

new CandidatePipelineBuilder[

Query,

CandidateSourceQuery,

CandidateSourceResult,

Result

](

queryFeatureHydratorExecutor,

asyncFeatureMapExecutor,

candidateDecoratorExecutor,

candidateFeatureHydratorExecutor,

candidateSourceExecutor,

groupResultsExecutor,

filterExecutor,

gateExecutor,

statsReceiver

)

}

}