package com.twitter.home\_mixer.candidate\_pipeline

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.NamesFeatureHydrator

import com.twitter.home\_mixer.functional\_component.feature\_hydrator.TweetypieFeatureHydrator

import com.twitter.home\_mixer.functional\_component.filter.InvalidSubscriptionTweetFilter

import com.twitter.product\_mixer.component\_library.candidate\_source.tweetconvosvc.ConversationServiceCandidateSource

import com.twitter.product\_mixer.component\_library.model.candidate.TweetCandidate

import com.twitter.product\_mixer.core.functional\_component.decorator.CandidateDecorator

import com.twitter.product\_mixer.core.functional\_component.gate.BaseGate

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ConversationServiceCandidatePipelineConfigBuilder[Query <: PipelineQuery] @Inject() (

conversationServiceCandidateSource: ConversationServiceCandidateSource,

tweetypieFeatureHydrator: TweetypieFeatureHydrator,

invalidSubscriptionTweetFilter: InvalidSubscriptionTweetFilter,

namesFeatureHydrator: NamesFeatureHydrator) {

def build(

gates: Seq[BaseGate[Query]] = Seq.empty,

decorator: Option[CandidateDecorator[Query, TweetCandidate]] = None

): ConversationServiceCandidatePipelineConfig[Query] = {

new ConversationServiceCandidatePipelineConfig(

conversationServiceCandidateSource,

tweetypieFeatureHydrator,

namesFeatureHydrator,

invalidSubscriptionTweetFilter,

gates,

decorator

)

}

}