package com.twitter.frigate.pushservice.take.predicates.candidate\_map

import com.twitter.frigate.pushservice.model.PushTypes.PushCandidate

import com.twitter.frigate.pushservice.model.\_

import com.twitter.frigate.pushservice.config.Config

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.frigate.thriftscala.CommonRecommendationType.\_

import com.twitter.hermit.predicate.NamedPredicate

object SendHandlerCandidatePredicatesMap {

def preCandidatePredicates(

implicit config: Config

): Map[CommonRecommendationType, List[NamedPredicate[\_ <: PushCandidate]]] = {

val magicFanoutNewsEventCandidatePredicates =

MagicFanoutNewsEventCandidatePredicates(config).preCandidateSpecificPredicates

val scheduledSpaceSubscriberPredicates = ScheduledSpaceSubscriberCandidatePredicates(

config).preCandidateSpecificPredicates

val scheduledSpaceSpeakerPredicates = ScheduledSpaceSpeakerCandidatePredicates(

config).preCandidateSpecificPredicates

val magicFanoutSportsEventCandidatePredicates =

MagicFanoutSportsEventCandidatePredicates(config).preCandidateSpecificPredicates

val magicFanoutProductLaunchPredicates = MagicFanoutProductLaunchPushCandidatePredicates(

config).preCandidateSpecificPredicates

val creatorSubscriptionFanoutPredicates = MagicFanouCreatorSubscriptionEventPushPredicates(

config).preCandidateSpecificPredicates

val newCreatorFanoutPredicates = MagicFanoutNewCreatorEventPushPredicates(

config).preCandidateSpecificPredicates

Map(

MagicFanoutNewsEvent -> magicFanoutNewsEventCandidatePredicates,

ScheduledSpaceSubscriber -> scheduledSpaceSubscriberPredicates,

ScheduledSpaceSpeaker -> scheduledSpaceSpeakerPredicates,

MagicFanoutSportsEvent -> magicFanoutSportsEventCandidatePredicates,

MagicFanoutProductLaunch -> magicFanoutProductLaunchPredicates,

NewCreator -> newCreatorFanoutPredicates,

CreatorSubscriber -> creatorSubscriptionFanoutPredicates

)

}

def postCandidatePredicates(

implicit config: Config

): Map[CommonRecommendationType, List[NamedPredicate[\_ <: PushCandidate]]] = {

val magicFanoutNewsEventCandidatePredicates =

MagicFanoutNewsEventCandidatePredicates(config).postCandidateSpecificPredicates

val scheduledSpaceSubscriberPredicates = ScheduledSpaceSubscriberCandidatePredicates(

config).postCandidateSpecificPredicates

val scheduledSpaceSpeakerPredicates = ScheduledSpaceSpeakerCandidatePredicates(

config).postCandidateSpecificPredicates

val magicFanoutSportsEventCandidatePredicates =

MagicFanoutSportsEventCandidatePredicates(config).postCandidateSpecificPredicates

val magicFanoutProductLaunchPredicates = MagicFanoutProductLaunchPushCandidatePredicates(

config).postCandidateSpecificPredicates

val creatorSubscriptionFanoutPredicates = MagicFanouCreatorSubscriptionEventPushPredicates(

config).postCandidateSpecificPredicates

val newCreatorFanoutPredicates = MagicFanoutNewCreatorEventPushPredicates(

config).postCandidateSpecificPredicates

Map(

MagicFanoutNewsEvent -> magicFanoutNewsEventCandidatePredicates,

ScheduledSpaceSubscriber -> scheduledSpaceSubscriberPredicates,

ScheduledSpaceSpeaker -> scheduledSpaceSpeakerPredicates,

MagicFanoutSportsEvent -> magicFanoutSportsEventCandidatePredicates,

MagicFanoutProductLaunch -> magicFanoutProductLaunchPredicates,

NewCreator -> newCreatorFanoutPredicates,

CreatorSubscriber -> creatorSubscriptionFanoutPredicates

)

}

}