package com.twitter.frigate.pushservice.send\_handler.generator

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

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

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

import com.twitter.frigate.pushservice.exception.UnsupportedCrtException

import com.twitter.frigate.thriftscala.FrigateNotification

import com.twitter.frigate.thriftscala.{CommonRecommendationType => CRT}

import com.twitter.util.Future

object PushRequestToCandidate {

final def generatePushCandidate(

frigateNotification: FrigateNotification,

target: Target

)(

implicit config: Config

): Future[RawCandidate] = {

val candidateGenerator: (Target, FrigateNotification) => Future[RawCandidate] = {

frigateNotification.commonRecommendationType match {

case CRT.MagicFanoutNewsEvent => MagicFanoutNewsEventCandidateGenerator.getCandidate

case CRT.ScheduledSpaceSubscriber => ScheduledSpaceSubscriberCandidateGenerator.getCandidate

case CRT.ScheduledSpaceSpeaker => ScheduledSpaceSpeakerCandidateGenerator.getCandidate

case CRT.MagicFanoutSportsEvent =>

MagicFanoutSportsEventCandidateGenerator.getCandidate(

\_,

\_,

config.basketballGameScoreStore,

config.baseballGameScoreStore,

config.cricketMatchScoreStore,

config.soccerMatchScoreStore,

config.nflGameScoreStore,

config.semanticCoreMegadataStore

)

case CRT.MagicFanoutProductLaunch =>

MagicFanoutProductLaunchCandidateGenerator.getCandidate

case CRT.NewCreator =>

MagicFanoutCreatorEventCandidateGenerator.getCandidate

case CRT.CreatorSubscriber =>

MagicFanoutCreatorEventCandidateGenerator.getCandidate

case \_ =>

throw new UnsupportedCrtException(

"UnsupportedCrtException for SendHandler: " + frigateNotification.commonRecommendationType)

}

}

candidateGenerator(target, frigateNotification)

}

}