package com.twitter.frigate.pushservice.take.candidate\_validator

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

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

import com.twitter.frigate.pushservice.take.predicates.candidate\_map.SendHandlerCandidatePredicatesMap

import com.twitter.hermit.predicate.Predicate

import com.twitter.util.Future

class SendHandlerPreCandidateValidator(override val config: Config) extends CandidateValidator {

override protected val candidatePredicatesMap =

SendHandlerCandidatePredicatesMap.preCandidatePredicates(config)

private val sendHandlerPreCandidateValidatorStats =

statsReceiver.counter("sendHandlerPreCandidateValidator\_stats")

override def validateCandidate[C <: PushCandidate](candidate: C): Future[Option[Predicate[C]]] = {

val candidatePredicates = getCRTPredicates(candidate.commonRecType)

val predicates = sendHandlerPrePredicates ++ candidatePredicates

sendHandlerPreCandidateValidatorStats.incr()

executeSequentialPredicates(candidate, predicates)

}

}