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.CandidatePredicatesMap

import com.twitter.hermit.predicate.Predicate

import com.twitter.util.Future

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

private val rFPHCandidateValidatorStats = statsReceiver.scope(this.getClass.getSimpleName)

private val concurrentPredicateCount = rFPHCandidateValidatorStats.counter("concurrent")

private val sequentialPredicateCount = rFPHCandidateValidatorStats.counter("sequential")

override protected val candidatePredicatesMap = CandidatePredicatesMap(config)

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

val candidatePredicates = getCRTPredicates(candidate.commonRecType)

val predicates = rfphPrePredicates ++ candidatePredicates ++ postPredicates

if (candidate.target.isEmailUser) {

concurrentPredicateCount.incr()

executeConcurrentPredicates(candidate, predicates).map(\_.headOption)

} else {

sequentialPredicateCount.incr()

executeSequentialPredicates(candidate, predicates)

}

}

}