package com.twitter.frigate.pushservice.take.predicates

import com.twitter.finagle.stats.StatsReceiver

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

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

import com.twitter.frigate.pushservice.predicate.CrtDeciderPredicate

import com.twitter.frigate.pushservice.predicate.PredicatesForCandidate

import com.twitter.frigate.pushservice.predicate.ScarecrowPredicate

import com.twitter.frigate.pushservice.predicate.ntab\_caret\_fatigue.NtabCaretClickFatiguePredicate

import com.twitter.hermit.predicate.NamedPredicate

trait TakeCommonPredicates {

def config: Config

implicit def statsReceiver: StatsReceiver

lazy val rfphPrePredicates: List[NamedPredicate[PushCandidate]] = List(

CrtDeciderPredicate(config.decider),

PredicatesForCandidate.isChannelValidPredicate,

)

lazy val sendHandlerPrePredicates: List[NamedPredicate[PushCandidate]] = List(

CrtDeciderPredicate(config.decider),

PredicatesForCandidate.enableSendHandlerCandidates,

PredicatesForCandidate.mrWebHoldbackPredicate,

PredicatesForCandidate.targetUserExists,

PredicatesForCandidate.authorInSocialContext,

PredicatesForCandidate.recommendedTweetIsAuthoredBySelf,

PredicatesForCandidate.selfInSocialContext,

NtabCaretClickFatiguePredicate()

)

lazy val postPredicates: List[NamedPredicate[PushCandidate]] = List(

ScarecrowPredicate(config.scarecrowCheckEventStore)

)

}