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

import com.twitter.frigate.common.base.TweetCandidate

import com.twitter.frigate.common.base.TweetDetails

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

import com.twitter.hermit.predicate.NamedPredicate

trait BasicTweetPredicatesForRFPH[C <: PushCandidate with TweetCandidate with TweetDetails]

extends BasicTweetPredicates

with BasicRFPHPredicates[C] {

// specific predicates per candidate type before basic tweet predicates

def preCandidateSpecificPredicates: List[NamedPredicate[C]] = List.empty

// specific predicates per candidate type after basic tweet predicates

def postCandidateSpecificPredicates: List[NamedPredicate[C]] = List.empty

override lazy val predicates: List[NamedPredicate[C]] =

preCandidateSpecificPredicates ++ basicTweetPredicates ++ postCandidateSpecificPredicates

}

/\*\*

\* This trait is a new version of BasicTweetPredicatesForRFPH

\* Difference from old version is that basicTweetPredicates are different

\* basicTweetPredicates here don't include Social Graph Service related predicates

\*/

trait BasicTweetPredicatesForRFPHWithoutSGSPredicates[

C <: PushCandidate with TweetCandidate with TweetDetails]

extends BasicTweetPredicatesWithoutSGSPredicates

with BasicRFPHPredicates[C] {

// specific predicates per candidate type before basic tweet predicates

def preCandidateSpecificPredicates: List[NamedPredicate[C]] = List.empty

// specific predicates per candidate type after basic tweet predicates

def postCandidateSpecificPredicates: List[NamedPredicate[C]] = List.empty

override lazy val predicates: List[NamedPredicate[C]] =

preCandidateSpecificPredicates ++ basicTweetPredicates ++ postCandidateSpecificPredicates

}