package com.twitter.frigate.pushservice.rank

import com.twitter.frigate.common.base.CandidateDetails

import com.twitter.frigate.common.base.Ranker

import com.twitter.util.Future

trait PushserviceRanker[T, C] extends Ranker[T, C] {

/\*\*

\* Initial Ranking of input candidates

\*/

def initialRank(target: T, candidates: Seq[CandidateDetails[C]]): Future[Seq[CandidateDetails[C]]]

/\*\*

\* Re-ranks input ranked candidates. Useful when a subset of candidates are ranked

\* by a different logic, while preserving the initial ranking for the rest

\*/

def reRank(

target: T,

rankedCandidates: Seq[CandidateDetails[C]]

): Future[Seq[CandidateDetails[C]]]

/\*\*

\* Final ranking that does Initial + Rerank

\*/

override final def rank(target: T, candidates: Seq[CandidateDetails[C]]): (

Future[Seq[CandidateDetails[C]]]

) = {

initialRank(target, candidates).flatMap { rankedCandidates => reRank(target, rankedCandidates) }

}

}