package com.twitter.product\_mixer.component\_library.selector

import com.twitter.product\_mixer.component\_library.model.candidate.RelevancePromptCandidate

import com.twitter.product\_mixer.core.model.common.presentation.CandidateWithDetails

import com.twitter.product\_mixer.core.model.common.presentation.ItemCandidateWithDetails

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

/\*\*

\* Compute a position for inserting a specific candidate into the result sequence originally provided to the Selector.

\* If a `None` is returned, the Selector using this would not insert that candidate into the result.

\*/

trait CandidatePositionInResults[-Query <: PipelineQuery] {

def apply(

query: Query,

candidate: CandidateWithDetails,

result: Seq[CandidateWithDetails]

): Option[Int]

}

object PromptCandidatePositionInResults extends CandidatePositionInResults[PipelineQuery] {

override def apply(

query: PipelineQuery,

candidate: CandidateWithDetails,

result: Seq[CandidateWithDetails]

): Option[Int] = candidate match {

case ItemCandidateWithDetails(candidate, \_, \_) =>

candidate match {

case relevancePromptCandidate: RelevancePromptCandidate => relevancePromptCandidate.position

case \_ => None

}

// not supporting ModuleCandidateWithDetails right now as RelevancePromptCandidate shouldn't be in a module

case \_ => None

}

}