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

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.functional\_component.common.CandidateScope

import com.twitter.product\_mixer.core.functional\_component.selector.Selector

import com.twitter.product\_mixer.core.functional\_component.selector.SelectorResult

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

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

/\*\*

\* A selector that appends all candidates missing a specific feature to the results pool and keeps

\* the rest in the remaining candidates. This is useful for backfill scoring candidates without

\* a score from a previous scorer.

\* @param pipelineScope The pipeline scope to check

\* @param missingFeature The missing feature to check for.

\*/

case class InsertAppendWithoutFeatureResults(

override val pipelineScope: CandidateScope,

missingFeature: Feature[\_, \_])

extends Selector[PipelineQuery] {

override def apply(

query: PipelineQuery,

remainingCandidates: Seq[CandidateWithDetails],

result: Seq[CandidateWithDetails]

): SelectorResult = {

val (candidatesWithMissingFeature, candidatesWithFeature) = remainingCandidates.partition {

candidate =>

pipelineScope.contains(candidate) && !candidate.features.getSuccessfulFeatures

.contains(missingFeature)

}

val updatedResults = result ++ candidatesWithMissingFeature

SelectorResult(remainingCandidates = candidatesWithFeature, result = updatedResults)

}

}