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

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

import CandidateScope.PartitionedCandidates

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

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

private[selector] object InsertSelector {

/\*\*

\* Insert all candidates from a candidate pipeline at a 0-indexed fixed position. If the current

\* results are a shorter length than the requested position, then the candidates will be appended

\* to the results.

\*/

def insertIntoResultsAtPosition(

position: Int,

pipelineScope: CandidateScope,

remainingCandidates: Seq[CandidateWithDetails],

result: Seq[CandidateWithDetails]

): SelectorResult = {

assert(position >= 0, "Position must be equal to or greater than zero")

val PartitionedCandidates(selectedCandidates, otherCandidates) =

pipelineScope.partition(remainingCandidates)

val resultUpdated = if (selectedCandidates.nonEmpty) {

if (position < result.length) {

val (left, right) = result.splitAt(position)

left ++ selectedCandidates ++ right

} else {

result ++ selectedCandidates

}

} else {

result

}

SelectorResult(remainingCandidates = otherCandidates, result = resultUpdated)

}

}