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

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

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

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

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.identifier.CandidatePipelineIdentifier

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

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

import com.twitter.timelines.configapi.Param

object InsertFixedPositionResults {

def apply(

candidatePipeline: CandidatePipelineIdentifier,

positionParam: Param[Int],

): InsertFixedPositionResults =

new InsertFixedPositionResults(SpecificPipeline(candidatePipeline), positionParam)

def apply(

candidatePipelines: Set[CandidatePipelineIdentifier],

positionParam: Param[Int]

): InsertFixedPositionResults =

new InsertFixedPositionResults(SpecificPipelines(candidatePipelines), positionParam)

}

/\*\*

\* Insert all candidates in a pipeline scope 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.

\*/

case class InsertFixedPositionResults(

override val pipelineScope: CandidateScope,

positionParam: Param[Int])

extends Selector[PipelineQuery] {

override def apply(

query: PipelineQuery,

remainingCandidates: Seq[CandidateWithDetails],

result: Seq[CandidateWithDetails]

): SelectorResult = InsertSelector.insertIntoResultsAtPosition(

position = query.params(positionParam),

pipelineScope = pipelineScope,

remainingCandidates = remainingCandidates,

result = result)

}