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

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

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

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

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

/\*\*

\* Drops all Candidates on the `remainingCandidates` side which are in the [[pipelineScope]]

\*

\* This is typically used as a placeholder when templating out a new pipeline or

\* as a simple filter to drop candidates based only on the [[CandidateScope]]

\*/

case class DropAllCandidates(override val pipelineScope: CandidateScope = AllPipelines)

extends Selector[PipelineQuery] {

override def apply(

query: PipelineQuery,

remainingCandidates: Seq[CandidateWithDetails],

result: Seq[CandidateWithDetails]

): SelectorResult = {

val PartitionedCandidates(inScope, outOfScope) = pipelineScope.partition(remainingCandidates)

SelectorResult(remainingCandidates = outOfScope, result = result)

}

}