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.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

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

import com.twitter.timelines.configapi.Param

/\*\*

\* Drop all results if the minimum item threshold is not met. Some products would rather return

\* nothing than, for example, a single tweet. This lets us leverage existing client logic for

\* handling no results such as logic to not render the product at all.

\*/

case class DropTooFewResults(minResultsParam: Param[Int]) extends Selector[PipelineQuery] {

override val pipelineScope: CandidateScope = AllPipelines

override def apply(

query: PipelineQuery,

remainingCandidates: Seq[CandidateWithDetails],

result: Seq[CandidateWithDetails]

): SelectorResult = {

val minResults = query.params(minResultsParam)

assert(minResults > 0, "Min results must be greater than zero")

if (PipelineResult.resultSize(result) < minResults) {

SelectorResult(remainingCandidates = remainingCandidates, result = Seq.empty)

} else {

SelectorResult(remainingCandidates = remainingCandidates, result = result)

}

}

}