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.timelines.configapi.Param

/\*\*

\* Limit the number of results

\*

\* For example, if maxResultsParam is 3, and the results contain 10 items, then these items will be

\* reduced to the first 3 selected items. Note that the ordering of results is determined by the

\* selector configuration.

\*

\* Another example, if maxResultsParam is 3, and the results contain 10 modules, then these will be

\* reduced to the first 3 modules. The items inside the modules will not be affected by this

\* selector.

\*/

case class DropMaxResults(

maxResultsParam: Param[Int])

extends Selector[PipelineQuery] {

override val pipelineScope: CandidateScope = AllPipelines

override def apply(

query: PipelineQuery,

remainingCandidates: Seq[CandidateWithDetails],

result: Seq[CandidateWithDetails]

): SelectorResult = {

val maxResults = query.params(maxResultsParam)

assert(maxResults > 0, "Max results must be greater than zero")

val resultUpdated = DropSelector.takeUntil(maxResults, result)

SelectorResult(remainingCandidates = remainingCandidates, result = resultUpdated)

}

}