package com.twitter.product\_mixer.component\_library.pipeline.candidate.ads

import com.twitter.product\_mixer.component\_library.model.query.ads.AdsQuery

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

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

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

/\*\*

\* Derive an estimate of the number of organic items from the query. If you need a more precise number,

\* consider switching to [[AdsDependentCandidatePipelineConfig]]

\*/

trait EstimateNumOrganicItems[Query <: PipelineQuery with AdsQuery] {

def apply(query: Query): Short

}

/\*\*

\* Compute the number of organic items from the query and set of previous candidates.

\*

\* @note the key difference between [[CountNumOrganicItems]] and [[EstimateNumOrganicItems]] is

\* that for [[EstimateNumOrganicItems]] we don't have any candidates returned yet, so we can

\* only guess as to the number of organic items in the result set. In contrast,

\* [[CountNumOrganicItems]] is used on dependant candidate pipelines where we can scan over

\* the candidate pipelines result set to count the number of organic items.

\*/

trait CountNumOrganicItems[-Query <: PipelineQuery with AdsQuery] {

def apply(query: Query, previousCandidates: Seq[CandidateWithDetails]): Short

}

/\*\*

\* Treat all previously retrieved candidates as organic

\*/

case object CountAllCandidates extends CountNumOrganicItems[PipelineQuery with AdsQuery] {

def apply(

query: PipelineQuery with AdsQuery,

previousCandidates: Seq[CandidateWithDetails]

): Short =

previousCandidates.length.toShort

}

/\*\*

\* Only count candidates from a specific subset of pipelines as organic

\*/

case class CountCandidatesFromPipelines(pipelines: CandidateScope)

extends CountNumOrganicItems[PipelineQuery with AdsQuery] {

def apply(

query: PipelineQuery with AdsQuery,

previousCandidates: Seq[CandidateWithDetails]

): Short =

previousCandidates.count(pipelines.contains).toShort

}