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

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

import com.twitter.product\_mixer.core.functional\_component.gate.QueryAndCandidateGate

import com.twitter.product\_mixer.core.model.common.identifier.GateIdentifier

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

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

import com.twitter.stitch.Stitch

/\*\*

\* A Gate that only continues if the previously returned candidates are empty. This is useful

\* for gating dependent candidate pipelines that are intedned to be used as a backfill when there

\* are no candidates available.

\*/

case class NoCandidatesGate(scope: CandidateScope) extends QueryAndCandidateGate[PipelineQuery] {

override val identifier: GateIdentifier = GateIdentifier("NoCandidates")

override def shouldContinue(

query: PipelineQuery,

candidates: Seq[CandidateWithDetails]

): Stitch[Boolean] = Stitch.value(scope.partition(candidates).candidatesInScope.isEmpty)

}