package com.twitter.product\_mixer.core.functional\_component.candidate\_source

import com.twitter.product\_mixer.core.model.common.Component

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

import com.twitter.stitch.Stitch

sealed trait BaseCandidateSource[-Request, +Candidate] extends Component {

/\*\* @see [[CandidateSourceIdentifier]] \*/

val identifier: CandidateSourceIdentifier

}

/\*\*

\* A [[CandidateSource]] returns a Seq of ''potential'' content

\*

\* @note [[CandidateSource]]s that return a single value need to transform

\* it into a Seq, either by doing `Seq(value)` or extracting

\* candidates from the value.

\*

\* @tparam Request arguments to get the potential content

\* @tparam Candidate the potential content

\*/

trait CandidateSource[-Request, +Candidate] extends BaseCandidateSource[Request, Candidate] {

/\*\* returns a Seq of ''potential'' content \*/

def apply(request: Request): Stitch[Seq[Candidate]]

}

/\*\*

\* A [[CandidateSourceWithExtractedFeatures]] returns a result containing both a Seq of

\* ''potential'' candidates as well as an extracted feature map that will later be appended

\* to the pipeline's [[com.twitter.product\_mixer.core.pipeline.PipelineQuery]] feature map. This is

\* useful for candidate sources that return features that might be useful later on without needing

\* to re-hydrate them.

\*

\* @note [[CandidateSource]]s that return a single value need to transform

\* it into a Seq, either by doing `Seq(value)` or extracting

\* candidates from the value.

\*

\* @tparam Request arguments to get the potential content

\* @tparam Candidate the potential content

\*/

trait CandidateSourceWithExtractedFeatures[-Request, +Candidate]

extends BaseCandidateSource[Request, Candidate] {

/\*\* returns a result containing a seq of ''potential'' content and extracted features

\* from the candidate source.

\* \*/

def apply(request: Request): Stitch[CandidatesWithSourceFeatures[Candidate]]

}