package com.twitter.product\_mixer.component\_library.candidate\_source.interest\_discovery

import com.google.inject.Inject

import com.google.inject.Singleton

import com.twitter.inject.Logging

import com.twitter.interests\_discovery.{thriftscala => t}

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

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

import com.twitter.stitch.Stitch

/\*\*

\* Generate a list of related topics results from IDS getRelatedTopics (thrift) endpoint.

\* Returns related topics, given a topic, whereas [[RecommendedTopicsCandidateSource]] returns

\* recommended topics, given a user.

\*/

@Singleton

class RelatedTopicsCandidateSource @Inject() (

interestDiscoveryService: t.InterestsDiscoveryService.MethodPerEndpoint)

extends CandidateSource[t.RelatedTopicsRequest, t.RelatedTopic]

with Logging {

override val identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier(name = "RelatedTopics")

override def apply(

request: t.RelatedTopicsRequest

): Stitch[Seq[t.RelatedTopic]] = {

Stitch

.callFuture(interestDiscoveryService.getRelatedTopics(request))

.map { response: t.RelatedTopicsResponse =>

response.topics

}

}

}