package com.twitter.product\_mixer.component\_library.candidate\_source.account\_recommendations\_mixer

import com.twitter.account\_recommendations\_mixer.{thriftscala => t}

import com.twitter.product\_mixer.component\_library.model.candidate.UserCandidate

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMapBuilder

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

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

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

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

object WhoToFollowModuleHeaderFeature extends Feature[UserCandidate, t.Header]

object WhoToFollowModuleFooterFeature extends Feature[UserCandidate, Option[t.Footer]]

object WhoToFollowModuleDisplayOptionsFeature

extends Feature[UserCandidate, Option[t.DisplayOptions]]

@Singleton

class AccountRecommendationsMixerCandidateSource @Inject() (

accountRecommendationsMixer: t.AccountRecommendationsMixer.MethodPerEndpoint)

extends CandidateSourceWithExtractedFeatures[

t.AccountRecommendationsMixerRequest,

t.RecommendedUser

] {

override val identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier(name = "AccountRecommendationsMixer")

override def apply(

request: t.AccountRecommendationsMixerRequest

): Stitch[CandidatesWithSourceFeatures[t.RecommendedUser]] = {

Stitch

.callFuture(accountRecommendationsMixer.getWtfRecommendations(request))

.map { response: t.WhoToFollowResponse =>

responseToCandidatesWithSourceFeatures(

response.userRecommendations,

response.header,

response.footer,

response.displayOptions)

}

}

private def responseToCandidatesWithSourceFeatures(

userRecommendations: Seq[t.RecommendedUser],

header: t.Header,

footer: Option[t.Footer],

displayOptions: Option[t.DisplayOptions],

): CandidatesWithSourceFeatures[t.RecommendedUser] = {

val features = FeatureMapBuilder()

.add(WhoToFollowModuleHeaderFeature, header)

.add(WhoToFollowModuleFooterFeature, footer)

.add(WhoToFollowModuleDisplayOptionsFeature, displayOptions)

.build()

CandidatesWithSourceFeatures(userRecommendations, features)

}

}