package com.twitter.product\_mixer.component\_library.candidate\_source.business\_profiles

import com.twitter.product\_mixer.component\_library.model.cursor.NextCursorFeature

import com.twitter.product\_mixer.component\_library.model.cursor.PreviousCursorFeature

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

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

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

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

import com.twitter.strato.client.Fetcher

import com.twitter.strato.generated.client.consumer\_identity.business\_profiles.BusinessProfileTeamMembersOnUserClientColumn

import com.twitter.strato.generated.client.consumer\_identity.business\_profiles.BusinessProfileTeamMembersOnUserClientColumn.{

Value => TeamMembersSlice

}

import com.twitter.strato.generated.client.consumer\_identity.business\_profiles.BusinessProfileTeamMembersOnUserClientColumn.{

View => TeamMembersView

}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class TeamMembersCandidateSource @Inject() (

column: BusinessProfileTeamMembersOnUserClientColumn)

extends StratoKeyViewFetcherWithSourceFeaturesSource[

Long,

TeamMembersView,

TeamMembersSlice,

Long

] {

override val identifier: CandidateSourceIdentifier = CandidateSourceIdentifier(

"BusinessProfileTeamMembers")

override val fetcher: Fetcher[Long, TeamMembersView, TeamMembersSlice] = column.fetcher

override def stratoResultTransformer(

stratoKey: Long,

stratoResult: TeamMembersSlice

): Seq[Long] =

stratoResult.members

override protected def extractFeaturesFromStratoResult(

stratoKey: Long,

stratoResult: TeamMembersSlice

): FeatureMap = {

val featureMapBuilder = FeatureMapBuilder()

stratoResult.previousCursor.foreach { cursor =>

featureMapBuilder.add(PreviousCursorFeature, cursor.toString)

}

stratoResult.nextCursor.foreach { cursor =>

featureMapBuilder.add(NextCursorFeature, cursor.toString)

}

featureMapBuilder.build()

}

}