package com.twitter.product\_mixer.component\_library.decorator.urt.builder.social\_context

import com.twitter.hermit.{thriftscala => h}

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.FeatureMap

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.social\_context.BaseSocialContextBuilder

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.FollowGeneralContextType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.GeneralContext

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.GeneralContextType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.LocationGeneralContextType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.NewUserGeneralContextType

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

case class WhoToFollowSocialContextBuilder(

socialTextFeature: Feature[\_, Option[String]],

contextTypeFeature: Feature[\_, Option[h.ContextType]])

extends BaseSocialContextBuilder[PipelineQuery, UserCandidate] {

def apply(

query: PipelineQuery,

candidate: UserCandidate,

candidateFeatures: FeatureMap

): Option[GeneralContext] = {

val socialTextOpt = candidateFeatures.getOrElse(socialTextFeature, None)

val contextTypeOpt = convertContextType(candidateFeatures.getOrElse(contextTypeFeature, None))

(socialTextOpt, contextTypeOpt) match {

case (Some(socialText), Some(contextType)) if socialText.nonEmpty =>

Some(

GeneralContext(

text = socialText,

contextType = contextType,

url = None,

contextImageUrls = None,

landingUrl = None))

case \_ => None

}

}

private def convertContextType(contextType: Option[h.ContextType]): Option[GeneralContextType] =

contextType match {

case Some(h.ContextType.Geo) => Some(LocationGeneralContextType)

case Some(h.ContextType.Social) => Some(FollowGeneralContextType)

case Some(h.ContextType.NewUser) => Some(NewUserGeneralContextType)

case \_ => None

}

}