package com.twitter.home\_mixer.functional\_component.decorator.urt.builder

import com.twitter.home\_mixer.model.HomeFeatures.SuggestTypeFeature

import com.twitter.home\_mixer.product.following.model.HomeMixerExternalStrings

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

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.\_

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

import com.twitter.product\_mixer.core.product.guice.scope.ProductScoped

import com.twitter.stringcenter.client.StringCenter

import com.twitter.timelineservice.suggests.{thriftscala => st}

import javax.inject.Inject

import javax.inject.Provider

import javax.inject.Singleton

@Singleton

case class PopularInYourAreaSocialContextBuilder @Inject() (

externalStrings: HomeMixerExternalStrings,

@ProductScoped stringCenterProvider: Provider[StringCenter])

extends BaseSocialContextBuilder[PipelineQuery, TweetCandidate] {

private val stringCenter = stringCenterProvider.get()

private val popularInYourAreaString = externalStrings.socialContextPopularInYourAreaString

def apply(

query: PipelineQuery,

candidate: TweetCandidate,

candidateFeatures: FeatureMap

): Option[SocialContext] = {

val suggestTypeOpt = candidateFeatures.getOrElse(SuggestTypeFeature, None)

if (suggestTypeOpt.contains(st.SuggestType.RecommendedTrendTweet)) {

Some(

GeneralContext(

contextType = LocationGeneralContextType,

text = stringCenter.prepare(popularInYourAreaString),

url = None,

contextImageUrls = None,

landingUrl = None

))

} else None

}

}