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

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

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

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.SocialContext

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 => t}

import javax.inject.Inject

import javax.inject.Provider

import javax.inject.Singleton

/\*\*

\* "Your Lists" will be rendered for the context and a url link for your lists.

\*/

@Singleton

case class ListsSocialContextBuilder @Inject() (

externalStrings: HomeMixerExternalStrings,

@ProductScoped stringCenterProvider: Provider[StringCenter])

extends BaseSocialContextBuilder[PipelineQuery, TweetCandidate] {

private val stringCenter = stringCenterProvider.get()

private val listString = externalStrings.ownedSubscribedListsModuleHeaderString

def apply(

query: PipelineQuery,

candidate: TweetCandidate,

candidateFeatures: FeatureMap

): Option[SocialContext] = {

candidateFeatures.get(SuggestTypeFeature) match {

case Some(suggestType) if suggestType == t.SuggestType.RankedListTweet =>

val userName = query.features.flatMap(\_.getOrElse(UserScreenNameFeature, None))

Some(

GeneralContext(

contextType = ListGeneralContextType,

text = stringCenter.prepare(listString),

url = userName.map(name => ""),

contextImageUrls = None,

landingUrl = None

))

case \_ => None

}

}

}