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

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

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

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 javax.inject.Inject

import javax.inject.Provider

import javax.inject.Singleton

@Singleton

case class FollowedBySocialContextBuilder @Inject() (

externalStrings: HomeMixerExternalStrings,

@ProductScoped stringCenterProvider: Provider[StringCenter])

extends BaseSocialContextBuilder[PipelineQuery, TweetCandidate] {

private val stringCenter = stringCenterProvider.get()

private val engagerSocialContextBuilder = EngagerSocialContextBuilder(

contextType = FollowGeneralContextType,

stringCenter = stringCenter,

oneUserString = externalStrings.socialContextOneUserFollowsString,

twoUsersString = externalStrings.socialContextTwoUsersFollowString,

moreUsersString = externalStrings.socialContextMoreUsersFollowString,

timelineTitle = externalStrings.socialContextFollowedByTimelineTitle

)

def apply(

query: PipelineQuery,

candidate: TweetCandidate,

candidateFeatures: FeatureMap

): Option[SocialContext] = {

// Only apply followed-by social context for OON Tweets

val inNetwork = candidateFeatures.getOrElse(InNetworkFeature, true)

if (!inNetwork) {

val validFollowedByUserIds =

candidateFeatures.getOrElse(SGSValidFollowedByUserIdsFeature, Nil)

engagerSocialContextBuilder(

socialContextIds = validFollowedByUserIds,

query = query,

candidateFeatures = candidateFeatures

)

} else {

None

}

}

}