package com.twitter.home\_mixer.product.list\_tweets.decorator

import com.twitter.home\_mixer.functional\_component.decorator.builder.HomeConversationModuleMetadataBuilder

import com.twitter.home\_mixer.functional\_component.decorator.builder.ListClientEventDetailsBuilder

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

import com.twitter.product\_mixer.component\_library.decorator.urt.UrtItemCandidateDecorator

import com.twitter.product\_mixer.component\_library.decorator.urt.UrtMultipleModulesDecorator

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.tweet.TweetCandidateUrtItemBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.metadata.ClientEventInfoBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.StaticModuleDisplayTypeBuilder

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.TimelineModuleBuilder

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.timeline\_module.VerticalConversation

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

import com.twitter.timelines.injection.scribe.InjectionScribeUtil

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

object ListConversationServiceCandidateDecorator {

private val ConversationModuleNamespace = EntryNamespace("list-conversation")

def apply(): Some[UrtMultipleModulesDecorator[PipelineQuery, TweetCandidate, Long]] = {

val suggestType = st.SuggestType.OrganicListTweet

val component = InjectionScribeUtil.scribeComponent(suggestType).get

val clientEventInfoBuilder = ClientEventInfoBuilder(

component = component,

detailsBuilder = Some(ListClientEventDetailsBuilder(st.SuggestType.OrganicListTweet))

)

val tweetItemBuilder = TweetCandidateUrtItemBuilder(

clientEventInfoBuilder = clientEventInfoBuilder

)

val moduleBuilder = TimelineModuleBuilder(

entryNamespace = ConversationModuleNamespace,

clientEventInfoBuilder = clientEventInfoBuilder,

displayTypeBuilder = StaticModuleDisplayTypeBuilder(VerticalConversation),

metadataBuilder = Some(HomeConversationModuleMetadataBuilder())

)

Some(

UrtMultipleModulesDecorator(

urtItemCandidateDecorator = UrtItemCandidateDecorator(tweetItemBuilder),

moduleBuilder = moduleBuilder,

groupByKey = (\_, \_, candidateFeatures) =>

candidateFeatures.getOrElse(ConversationModuleFocalTweetIdFeature, None)

))

}

}