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

import com.twitter.product\_mixer.core.functional\_component.configapi.StaticParam

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseModuleDisplayTypeBuilder

import com.twitter.product\_mixer.core.model.common.CandidateWithFeatures

import com.twitter.product\_mixer.core.model.common.UniversalNoun

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

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

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

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

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

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

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

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

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

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

import com.twitter.timelines.configapi.Param

object WhoToFollowModuleDisplayType extends Enumeration {

type ModuleDisplayType = Value

val Carousel = Value

val CompactCarousel = Value

val ConversationTree = Value

val GridCarousel = Value

val Vertical = Value

val VerticalConversation = Value

val VerticalGrid = Value

val VerticalWithContextLine = Value

}

case class ParamWhoToFollowModuleDisplayTypeBuilder(

displayTypeParam: Param[WhoToFollowModuleDisplayType.Value] =

StaticParam(WhoToFollowModuleDisplayType.Vertical))

extends BaseModuleDisplayTypeBuilder[PipelineQuery, UniversalNoun[Any]] {

override def apply(

query: PipelineQuery,

candidates: Seq[CandidateWithFeatures[UniversalNoun[Any]]]

): ModuleDisplayType = {

val displayType = query.params(displayTypeParam)

displayType match {

case WhoToFollowModuleDisplayType.Carousel => Carousel

case WhoToFollowModuleDisplayType.CompactCarousel => CompactCarousel

case WhoToFollowModuleDisplayType.ConversationTree => ConversationTree

case WhoToFollowModuleDisplayType.GridCarousel => GridCarousel

case WhoToFollowModuleDisplayType.Vertical => Vertical

case WhoToFollowModuleDisplayType.VerticalConversation => VerticalConversation

case WhoToFollowModuleDisplayType.VerticalGrid => VerticalGrid

case WhoToFollowModuleDisplayType.VerticalWithContextLine => VerticalWithContextLine

}

}

}