package com.twitter.home\_mixer.product.subscribed.model

import com.twitter.home\_mixer.model.request.DeviceContext

import com.twitter.home\_mixer.model.request.HasDeviceContext

import com.twitter.home\_mixer.model.request.HasSeenTweetIds

import com.twitter.home\_mixer.model.request.SubscribedProduct

import com.twitter.product\_mixer.component\_library.model.cursor.UrtOrderedCursor

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.model.marshalling.request.\_

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

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

import com.twitter.timelines.configapi.Params

case class SubscribedQuery(

override val params: Params,

override val clientContext: ClientContext,

override val pipelineCursor: Option[UrtOrderedCursor],

override val requestedMaxResults: Option[Int],

override val debugOptions: Option[DebugOptions],

override val features: Option[FeatureMap],

override val deviceContext: Option[DeviceContext],

override val seenTweetIds: Option[Seq[Long]])

extends PipelineQuery

with HasPipelineCursor[UrtOrderedCursor]

with HasDeviceContext

with HasSeenTweetIds {

override val product: Product = SubscribedProduct

override def withFeatureMap(features: FeatureMap): SubscribedQuery =

copy(features = Some(features))

}