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

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

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.GridCarouselMetadata

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

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

import com.twitter.timelines.configapi.Param

object TopicsToFollowModuleMetadataBuilder {

val TopicsPerRow = 7

/\*

\* rows = min(MAX\_NUM\_ROWS, # topics / TOPICS\_PER\_ROW)

\* where TOPICS\_PER\_ROW = 7

\*/

def getCarouselRowCount(topicsCount: Int, maxCarouselRows: Int): Int =

Math.min(maxCarouselRows, (topicsCount / TopicsPerRow) + 1)

}

case class TopicsToFollowModuleMetadataBuilder(maxCarouselRowsParam: Param[Int])

extends BaseModuleMetadataBuilder[PipelineQuery, UniversalNoun[Any]] {

import TopicsToFollowModuleMetadataBuilder.\_

override def apply(

query: PipelineQuery,

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

): ModuleMetadata = {

val rowCount = getCarouselRowCount(candidates.size, query.params(maxCarouselRowsParam))

ModuleMetadata(

adsMetadata = None,

conversationMetadata = None,

gridCarouselMetadata = Some(GridCarouselMetadata(numRows = Some(rowCount)))

)

}

}