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

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.topic.BasicTopicFunctionalityType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.topic.PivotTopicFunctionalityType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.topic.RecommendationTopicFunctionalityType

import com.twitter.product\_mixer.core.model.marshalling.response.urt.item.topic.TopicFunctionalityType

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

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.item.topic.BaseTopicFunctionalityTypeBuilder

import com.twitter.timelines.configapi.FSEnumParam

object TopicFunctionalityTypeParamValue extends Enumeration {

type TopicFunctionalityType = Value

val Basic = Value

val Pivot = Value

val Recommendation = Value

}

case class ParamTopicFunctionalityTypeBuilder(

functionalityTypeParam: FSEnumParam[TopicFunctionalityTypeParamValue.type])

extends BaseTopicFunctionalityTypeBuilder[PipelineQuery, TopicCandidate] {

override def apply(

query: PipelineQuery,

candidate: TopicCandidate,

candidateFeatures: FeatureMap

): Option[TopicFunctionalityType] = {

val functionalityType = query.params(functionalityTypeParam)

functionalityType match {

case TopicFunctionalityTypeParamValue.Basic => Some(BasicTopicFunctionalityType)

case TopicFunctionalityTypeParamValue.Pivot => Some(PivotTopicFunctionalityType)

case TopicFunctionalityTypeParamValue.Recommendation =>

Some(RecommendationTopicFunctionalityType)

}

}

}