package com.twitter.follow\_recommendations.products.home\_timeline\_tweet\_recs

import com.twitter.follow\_recommendations.common.base.BaseRecommendationFlow

import com.twitter.follow\_recommendations.common.base.IdentityTransform

import com.twitter.follow\_recommendations.common.base.Transform

import com.twitter.follow\_recommendations.common.models.DisplayLocation

import com.twitter.follow\_recommendations.common.models.Recommendation

import com.twitter.follow\_recommendations.flows.content\_recommender\_flow.ContentRecommenderFlow

import com.twitter.follow\_recommendations.flows.content\_recommender\_flow.ContentRecommenderRequestBuilder

import com.twitter.follow\_recommendations.products.common.Product

import com.twitter.follow\_recommendations.products.common.ProductRequest

import com.twitter.follow\_recommendations.products.home\_timeline\_tweet\_recs.configapi.HomeTimelineTweetRecsParams.\_

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

/\*

\* This "DisplayLocation" is used to generate user recommendations using the ContentRecommenderFlow. These recommendations are later used downstream

\* to generate recommended tweets on Home Timeline.

\*/

@Singleton

class HomeTimelineTweetRecsProduct @Inject() (

contentRecommenderFlow: ContentRecommenderFlow,

contentRecommenderRequestBuilder: ContentRecommenderRequestBuilder)

extends Product {

override val name: String = "Home Timeline Tweet Recs"

override val identifier: String = "home-timeline-tweet-recs"

override val displayLocation: DisplayLocation = DisplayLocation.HomeTimelineTweetRecs

override def selectWorkflows(

request: ProductRequest

): Stitch[Seq[BaseRecommendationFlow[ProductRequest, \_ <: Recommendation]]] = {

contentRecommenderRequestBuilder.build(request).map { contentRecommenderRequest =>

Seq(contentRecommenderFlow.mapKey({ request: ProductRequest => contentRecommenderRequest }))

}

}

override val blender: Transform[ProductRequest, Recommendation] =

new IdentityTransform[ProductRequest, Recommendation]

override def resultsTransformer(

request: ProductRequest

): Stitch[Transform[ProductRequest, Recommendation]] =

Stitch.value(new IdentityTransform[ProductRequest, Recommendation])

override def enabled(request: ProductRequest): Stitch[Boolean] =

Stitch.value(request.params(EnableProduct))

}