package com.twitter.follow\_recommendations.products.explore\_tab

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.post\_nux\_ml.PostNuxMlFlow

import com.twitter.follow\_recommendations.flows.post\_nux\_ml.PostNuxMlRequestBuilder

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

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

import com.twitter.follow\_recommendations.products.explore\_tab.configapi.ExploreTabParams

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ExploreTabProduct @Inject() (

postNuxMlFlow: PostNuxMlFlow,

postNuxMlRequestBuilder: PostNuxMlRequestBuilder)

extends Product {

override val name: String = "Explore Tab"

override val identifier: String = "explore-tab"

override val displayLocation: DisplayLocation = DisplayLocation.ExploreTab

override def selectWorkflows(

request: ProductRequest

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

postNuxMlRequestBuilder.build(request).map { postNuxMlRequest =>

Seq(postNuxMlFlow.mapKey({ \_: ProductRequest => postNuxMlRequest }))

}

}

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] = {

// Ideally we should hook up is\_soft\_user as custom FS field and disable the product through FS

val enabledForUserType = !request.recommendationRequest.isSoftUser || request.params(

ExploreTabParams.EnableProductForSoftUser)

Stitch.value(request.params(ExploreTabParams.EnableProduct) && enabledForUserType)

}

}