package com.twitter.follow\_recommendations.controllers

import com.twitter.finatra.thrift.Controller

import com.twitter.follow\_recommendations.configapi.ParamsFactory

import com.twitter.follow\_recommendations.services.ProductPipelineSelector

import com.twitter.follow\_recommendations.services.UserScoringService

import com.twitter.follow\_recommendations.thriftscala.FollowRecommendationsThriftService

import com.twitter.follow\_recommendations.thriftscala.FollowRecommendationsThriftService.\_

import com.twitter.stitch.Stitch

import javax.inject.Inject

class ThriftController @Inject() (

userScoringService: UserScoringService,

recommendationRequestBuilder: RecommendationRequestBuilder,

scoringUserRequestBuilder: ScoringUserRequestBuilder,

productPipelineSelector: ProductPipelineSelector,

paramsFactory: ParamsFactory)

extends Controller(FollowRecommendationsThriftService) {

handle(GetRecommendations) { args: GetRecommendations.Args =>

val stitch = recommendationRequestBuilder.fromThrift(args.request).flatMap { request =>

val params = paramsFactory(

request.clientContext,

request.displayLocation,

request.debugParams.flatMap(\_.featureOverrides).getOrElse(Map.empty))

productPipelineSelector.selectPipeline(request, params).map(\_.toThrift)

}

Stitch.run(stitch)

}

handle(ScoreUserCandidates) { args: ScoreUserCandidates.Args =>

val stitch = scoringUserRequestBuilder.fromThrift(args.request).flatMap { request =>

val params = paramsFactory(

request.clientContext,

request.displayLocation,

request.debugParams.flatMap(\_.featureOverrides).getOrElse(Map.empty))

userScoringService.get(request.copy(params = params)).map(\_.toThrift)

}

Stitch.run(stitch)

}

}