package com.twitter.product\_mixer.component\_library.selector.ads

import com.google.inject.Inject

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.goldfinch.adaptors.ads.productmixer.ProductMixerPromotedEntriesAdaptor

import com.twitter.goldfinch.adaptors.productmixer.ProductMixerNonPromotedEntriesAdaptor

import com.twitter.goldfinch.adaptors.productmixer.ProductMixerQueryConverter

import com.twitter.goldfinch.api.AdsInjectionRequestContextConverter

import com.twitter.goldfinch.api.AdsInjectionSurfaceAreas.SurfaceAreaName

import com.twitter.goldfinch.api.{AdsInjector => GoldfinchAdsInjector}

import com.twitter.goldfinch.api.NonPromotedEntriesAdaptor

import com.twitter.goldfinch.api.PromotedEntriesAdaptor

import com.twitter.goldfinch.impl.injector.AdsInjectorBuilder

import com.twitter.goldfinch.impl.injector.product\_mixer.AdsInjectionSurfaceAreaAdjustersMap

import com.twitter.goldfinch.impl.injector.product\_mixer.VerticalSizeAdjustmentConfigMap

import com.twitter.inject.Logging

import com.twitter.product\_mixer.component\_library.model.query.ads.\_

import com.twitter.product\_mixer.core.model.common.presentation.\_

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

import javax.inject.Singleton

import com.twitter.goldfinch.impl.core.DefaultFeatureSwitchResultsFactory

import com.twitter.goldfinch.impl.core.LocalDevelopmentFeatureSwitchResultsFactory

import com.twitter.inject.annotations.Flag

import com.twitter.product\_mixer.core.module.product\_mixer\_flags.ProductMixerFlagModule.ConfigRepoLocalPath

import com.twitter.product\_mixer.core.module.product\_mixer\_flags.ProductMixerFlagModule.ServiceLocal

@Singleton

class AdsInjector @Inject() (

statsReceiver: StatsReceiver,

@Flag(ConfigRepoLocalPath) localConfigRepoPath: String,

@Flag(ServiceLocal) isServiceLocal: Boolean)

extends Logging {

private val adsQueryRequestConverter: AdsInjectionRequestContextConverter[

PipelineQuery with AdsQuery

] = ProductMixerQueryConverter

def forSurfaceArea(

surfaceAreaName: SurfaceAreaName

): GoldfinchAdsInjector[

PipelineQuery with AdsQuery,

CandidateWithDetails,

CandidateWithDetails

] = {

val scopedStatsReceiver: StatsReceiver =

statsReceiver.scope("goldfinch", surfaceAreaName.toString)

val nonAdsAdaptor: NonPromotedEntriesAdaptor[CandidateWithDetails] =

ProductMixerNonPromotedEntriesAdaptor(

VerticalSizeAdjustmentConfigMap.configsBySurfaceArea(surfaceAreaName),

scopedStatsReceiver)

val adsAdaptor: PromotedEntriesAdaptor[CandidateWithDetails] =

new ProductMixerPromotedEntriesAdaptor(scopedStatsReceiver)

val featureSwitchFactory = if (isServiceLocal) {

new LocalDevelopmentFeatureSwitchResultsFactory(

surfaceAreaName.toString,

configRepoAbsPath = localConfigRepoPath)

} else new DefaultFeatureSwitchResultsFactory(surfaceAreaName.toString)

new AdsInjectorBuilder[PipelineQuery with AdsQuery, CandidateWithDetails, CandidateWithDetails](

requestAdapter = adsQueryRequestConverter,

nonPromotedEntriesAdaptor = nonAdsAdaptor,

promotedEntriesAdaptor = adsAdaptor,

adjusters =

AdsInjectionSurfaceAreaAdjustersMap.getAdjusters(surfaceAreaName, scopedStatsReceiver),

featureSwitchFactory = featureSwitchFactory,

statsReceiver = scopedStatsReceiver,

logger = logger

).build()

}

}