package com.twitter.product\_mixer.component\_library.pipeline.candidate.ads

import com.twitter.adserver.thriftscala.AdImpression

import com.twitter.adserver.thriftscala.AdRequestParams

import com.twitter.product\_mixer.component\_library.decorator.urt.UrtItemCandidateDecorator

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.item.ad.AdsCandidateUrtItemBuilder

import com.twitter.product\_mixer.component\_library.model.candidate.ads.AdsCandidate

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

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

import com.twitter.product\_mixer.core.functional\_component.common.alert.Alert

import com.twitter.product\_mixer.core.functional\_component.decorator.CandidateDecorator

import com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.BaseCandidateFeatureHydrator

import com.twitter.product\_mixer.core.functional\_component.filter.Filter

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.model.common.identifier.CandidatePipelineIdentifier

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

import com.twitter.timelines.configapi.FSParam

import com.twitter.timelines.configapi.decider.DeciderParam

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class AdsCandidatePipelineConfigBuilder @Inject() () {

def build[Query <: PipelineQuery with AdsQuery](

adsCandidateSource: CandidateSource[AdRequestParams, AdImpression],

adsDisplayLocationBuilder: AdsDisplayLocationBuilder[Query],

estimateNumOrganicItems: EstimateNumOrganicItems[Query],

identifier: CandidatePipelineIdentifier = CandidatePipelineIdentifier("Ads"),

enabledDeciderParam: Option[DeciderParam[Boolean]] = None,

supportedClientParam: Option[FSParam[Boolean]] = None,

gates: Seq[Gate[Query]] = Seq.empty,

filters: Seq[Filter[Query, AdsCandidate]] = Seq.empty,

postFilterFeatureHydration: Seq[BaseCandidateFeatureHydrator[Query, AdsCandidate, \_]] =

Seq.empty,

decorator: Option[CandidateDecorator[Query, AdsCandidate]] =

Some(UrtItemCandidateDecorator(AdsCandidateUrtItemBuilder())),

alerts: Seq[Alert] = Seq.empty,

urtRequest: Option[Boolean] = None,

): AdsCandidatePipelineConfig[Query] = {

new AdsCandidatePipelineConfig(

identifier = identifier,

enabledDeciderParam = enabledDeciderParam,

supportedClientParam = supportedClientParam,

gates = gates,

candidateSource = adsCandidateSource,

filters = filters,

postFilterFeatureHydration = postFilterFeatureHydration,

decorator = decorator,

alerts = alerts,

adsDisplayLocationBuilder = adsDisplayLocationBuilder,

estimateNumOrganicItems = estimateNumOrganicItems,

urtRequest = urtRequest,

)

}

}