package com.twitter.frigate.pushservice.adaptor

import com.twitter.content\_mixer.thriftscala.ContentMixerRequest

import com.twitter.content\_mixer.thriftscala.ContentMixerResponse

import com.twitter.explore\_ranker.thriftscala.ExploreRankerRequest

import com.twitter.explore\_ranker.thriftscala.ExploreRankerResponse

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.\_

import com.twitter.frigate.common.candidate.\_

import com.twitter.frigate.common.store.RecentTweetsQuery

import com.twitter.frigate.common.store.interests.InterestsLookupRequestWithContext

import com.twitter.frigate.pushservice.model.PushTypes.RawCandidate

import com.twitter.frigate.pushservice.model.PushTypes.Target

import com.twitter.frigate.pushservice.params.PushFeatureSwitchParams

import com.twitter.frigate.pushservice.store.\_

import com.twitter.geoduck.common.thriftscala.Location

import com.twitter.geoduck.service.thriftscala.LocationResponse

import com.twitter.hermit.pop\_geo.thriftscala.PopTweetsInPlace

import com.twitter.hermit.predicate.socialgraph.RelationEdge

import com.twitter.hermit.store.tweetypie.UserTweet

import com.twitter.interests.thriftscala.UserInterests

import com.twitter.interests\_discovery.thriftscala.NonPersonalizedRecommendedLists

import com.twitter.interests\_discovery.thriftscala.RecommendedListsRequest

import com.twitter.interests\_discovery.thriftscala.RecommendedListsResponse

import com.twitter.recommendation.interests.discovery.core.model.InterestDomain

import com.twitter.stitch.tweetypie.TweetyPie.TweetyPieResult

import com.twitter.storehaus.ReadableStore

import com.twitter.trends.trip\_v1.trip\_tweets.thriftscala.TripDomain

import com.twitter.trends.trip\_v1.trip\_tweets.thriftscala.TripTweets

import com.twitter.tsp.thriftscala.TopicSocialProofRequest

import com.twitter.tsp.thriftscala.TopicSocialProofResponse

/\*\*

\* PushCandidateSourceGenerator generates candidate source list for a given Target user

\*/

class PushCandidateSourceGenerator(

earlybirdCandidates: CandidateSource[EarlybirdCandidateSource.Query, EarlybirdCandidate],

userTweetEntityGraphCandidates: CandidateSource[UserTweetEntityGraphCandidates.Target, Candidate],

cachedTweetyPieStoreV2: ReadableStore[Long, TweetyPieResult],

safeCachedTweetyPieStoreV2: ReadableStore[Long, TweetyPieResult],

userTweetTweetyPieStore: ReadableStore[UserTweet, TweetyPieResult],

safeUserTweetTweetyPieStore: ReadableStore[UserTweet, TweetyPieResult],

cachedTweetyPieStoreV2NoVF: ReadableStore[Long, TweetyPieResult],

edgeStore: ReadableStore[RelationEdge, Boolean],

interestsLookupStore: ReadableStore[InterestsLookupRequestWithContext, UserInterests],

uttEntityHydrationStore: UttEntityHydrationStore,

geoDuckV2Store: ReadableStore[Long, LocationResponse],

topTweetsByGeoStore: ReadableStore[InterestDomain[String], Map[String, List[(Long, Double)]]],

topTweetsByGeoV2VersionedStore: ReadableStore[String, PopTweetsInPlace],

tweetImpressionsStore: TweetImpressionsStore,

recommendedTrendsCandidateSource: RecommendedTrendsCandidateSource,

recentTweetsByAuthorStore: ReadableStore[RecentTweetsQuery, Seq[Seq[Long]]],

topicSocialProofServiceStore: ReadableStore[TopicSocialProofRequest, TopicSocialProofResponse],

crMixerStore: CrMixerTweetStore,

contentMixerStore: ReadableStore[ContentMixerRequest, ContentMixerResponse],

exploreRankerStore: ReadableStore[ExploreRankerRequest, ExploreRankerResponse],

softUserLocationStore: ReadableStore[Long, Location],

tripTweetCandidateStore: ReadableStore[TripDomain, TripTweets],

listRecsStore: ReadableStore[String, NonPersonalizedRecommendedLists],

idsStore: ReadableStore[RecommendedListsRequest, RecommendedListsResponse]

)(

implicit val globalStats: StatsReceiver) {

private val earlyBirdFirstDegreeCandidateAdaptor = EarlyBirdFirstDegreeCandidateAdaptor(

earlybirdCandidates,

cachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

userTweetTweetyPieStore,

PushFeatureSwitchParams.NumberOfMaxEarlybirdInNetworkCandidatesParam,

globalStats

)

private val frsTweetCandidateAdaptor = FRSTweetCandidateAdaptor(

crMixerStore,

cachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

userTweetTweetyPieStore,

uttEntityHydrationStore,

topicSocialProofServiceStore,

globalStats

)

private val contentRecommenderMixerAdaptor = ContentRecommenderMixerAdaptor(

crMixerStore,

safeCachedTweetyPieStoreV2,

edgeStore,

topicSocialProofServiceStore,

uttEntityHydrationStore,

globalStats

)

private val tripGeoCandidatesAdaptor = TripGeoCandidatesAdaptor(

tripTweetCandidateStore,

contentMixerStore,

safeCachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

globalStats

)

val sources: Seq[

CandidateSource[Target, RawCandidate] with CandidateSourceEligible[

Target,

RawCandidate

]

] = {

Seq(

earlyBirdFirstDegreeCandidateAdaptor,

GenericCandidateAdaptor(

userTweetEntityGraphCandidates,

cachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

globalStats.scope("UserTweetEntityGraphCandidates")

),

new OnboardingPushCandidateAdaptor(globalStats),

TopTweetsByGeoAdaptor(

geoDuckV2Store,

softUserLocationStore,

topTweetsByGeoStore,

topTweetsByGeoV2VersionedStore,

cachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

globalStats

),

frsTweetCandidateAdaptor,

TopTweetImpressionsCandidateAdaptor(

recentTweetsByAuthorStore,

cachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

tweetImpressionsStore,

globalStats

),

TrendsCandidatesAdaptor(

softUserLocationStore,

recommendedTrendsCandidateSource,

safeCachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

safeUserTweetTweetyPieStore,

globalStats

),

contentRecommenderMixerAdaptor,

tripGeoCandidatesAdaptor,

HighQualityTweetsAdaptor(

tripTweetCandidateStore,

interestsLookupStore,

cachedTweetyPieStoreV2,

cachedTweetyPieStoreV2NoVF,

globalStats

),

ExploreVideoTweetCandidateAdaptor(

exploreRankerStore,

cachedTweetyPieStoreV2,

globalStats

),

ListsToRecommendCandidateAdaptor(

listRecsStore,

geoDuckV2Store,

idsStore,

globalStats

)

)

}

}