package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.ScheduledSpaceSubscriberCandidate

import com.twitter.frigate.common.base.SpaceCandidateFanoutDetails

import com.twitter.frigate.common.util.FeatureSwitchParams

import com.twitter.frigate.magic\_events.thriftscala.SpaceMetadata

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

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

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

import com.twitter.frigate.pushservice.config.Config

import com.twitter.frigate.pushservice.ml.PushMLModelScorer

import com.twitter.frigate.pushservice.model.candidate.CopyIds

import com.twitter.frigate.pushservice.model.ibis.ScheduledSpaceSubscriberIbis2Hydrator

import com.twitter.frigate.pushservice.model.ntab.ScheduledSpaceSubscriberNTabRequestHydrator

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

import com.twitter.frigate.pushservice.take.predicates.BasicSendHandlerPredicates

import com.twitter.frigate.thriftscala.FrigateNotification

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.storehaus.ReadableStore

import com.twitter.ubs.thriftscala.AudioSpace

import com.twitter.util.Future

class ScheduledSpaceSubscriberPushCandidate(

candidate: RawCandidate with ScheduledSpaceSubscriberCandidate,

hostUser: Option[User],

copyIds: CopyIds,

audioSpaceStore: ReadableStore[String, AudioSpace]

)(

implicit val statsScoped: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with ScheduledSpaceSubscriberCandidate

with SpaceCandidateFanoutDetails

with ScheduledSpaceSubscriberIbis2Hydrator

with ScheduledSpaceSubscriberNTabRequestHydrator {

override val startTime: Long = candidate.startTime

override val hydratedHost: Option[User] = hostUser

override val spaceId: String = candidate.spaceId

override val hostId: Option[Long] = candidate.hostId

override val speakerIds: Option[Seq[Long]] = candidate.speakerIds

override val listenerIds: Option[Seq[Long]] = candidate.listenerIds

override val frigateNotification: FrigateNotification = candidate.frigateNotification

override val pushCopyId: Option[Int] = copyIds.pushCopyId

override val ntabCopyId: Option[Int] = copyIds.ntabCopyId

override val copyAggregationId: Option[String] = copyIds.aggregationId

override val target: Target = candidate.target

override lazy val audioSpaceFut: Future[Option[AudioSpace]] = audioSpaceStore.get(spaceId)

override val spaceFanoutMetadata: Option[SpaceMetadata] = None

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val statsReceiver: StatsReceiver =

statsScoped.scope("ScheduledSpaceSubscriberCandidate")

}

case class ScheduledSpaceSubscriberCandidatePredicates(config: Config)

extends BasicSendHandlerPredicates[ScheduledSpaceSubscriberPushCandidate] {

implicit val statsReceiver: StatsReceiver = config.statsReceiver.scope(getClass.getSimpleName)

override val preCandidateSpecificPredicates: List[

NamedPredicate[ScheduledSpaceSubscriberPushCandidate]

] =

List(

PredicatesForCandidate.paramPredicate(FeatureSwitchParams.EnableScheduledSpaceSubscribers),

SpacePredicate.narrowCastSpace,

SpacePredicate.targetInSpace(config.audioSpaceParticipantsStore),

SpacePredicate.spaceHostTargetUserBlocking(config.edgeStore),

PredicatesForCandidate.duplicateSpacesPredicate

)

}