package com.twitter.frigate.pushservice.send\_handler.generator

import com.twitter.frigate.common.base.ScheduledSpaceSpeakerCandidate

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

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

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.frigate.thriftscala.FrigateNotification

import com.twitter.util.Future

object ScheduledSpaceSpeakerCandidateGenerator extends CandidateGenerator {

override def getCandidate(

targetUser: Target,

notification: FrigateNotification

): Future[RawCandidate] = {

/\*\*

\* frigateNotification recommendation type should be [[CommonRecommendationType.ScheduledSpaceSpeaker]]

\*

\*\*/

require(

notification.commonRecommendationType == CommonRecommendationType.ScheduledSpaceSpeaker,

"ScheduledSpaceSpeaker: unexpected CRT " + notification.commonRecommendationType

)

val spaceNotification = notification.spaceNotification.getOrElse(

throw new IllegalStateException("ScheduledSpaceSpeaker notification object not defined"))

require(

spaceNotification.hostUserId.isDefined,

"ScheduledSpaceSpeaker notification - hostUserId not defined"

)

val spaceHostId = spaceNotification.hostUserId

require(

spaceNotification.scheduledStartTime.isDefined,

"ScheduledSpaceSpeaker notification - scheduledStartTime not defined"

)

val scheduledStartTime = spaceNotification.scheduledStartTime.get

val candidate = new RawCandidate with ScheduledSpaceSpeakerCandidate {

override val target: Target = targetUser

override val frigateNotification: FrigateNotification = notification

override val spaceId: String = spaceNotification.broadcastId

override val hostId: Option[Long] = spaceHostId

override val startTime: Long = scheduledStartTime

override val speakerIds: Option[Seq[Long]] = spaceNotification.speakers

override val listenerIds: Option[Seq[Long]] = spaceNotification.listeners

}

Future.value(candidate)

}

}