package com.twitter.frigate.pushservice.model.ibis

import com.twitter.frigate.pushservice.model.ScheduledSpaceSpeakerPushCandidate

import com.twitter.frigate.pushservice.util.PushIbisUtil.\_

import com.twitter.frigate.thriftscala.SpaceNotificationType

import com.twitter.util.Future

trait ScheduledSpaceSpeakerIbis2Hydrator extends Ibis2HydratorForCandidate {

self: ScheduledSpaceSpeakerPushCandidate =>

override lazy val senderId: Option[Long] = None

private lazy val targetModelValues: Future[Map[String, String]] = {

hostId match {

case Some(spaceHostId) =>

audioSpaceFut.map { audioSpace =>

val isStartNow = frigateNotification.spaceNotification.exists(

\_.spaceNotificationType.contains(SpaceNotificationType.AtSpaceBroadcast))

Map(

"host\_id" -> s"$spaceHostId",

"space\_id" -> spaceId,

"is\_start\_now" -> s"$isStartNow"

) ++ audioSpace.flatMap(\_.title.map("space\_title" -> \_))

}

case \_ =>

Future.exception(

new IllegalStateException("Unable to get host id for ScheduledSpaceSpeakerIbis2Hydrator"))

}

}

override lazy val modelValues: Future[Map[String, String]] =

mergeFutModelValues(super.modelValues, targetModelValues)

}