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

import com.twitter.frigate.pushservice.model.ScheduledSpaceSubscriberPushCandidate

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

import com.twitter.util.Future

trait ScheduledSpaceSubscriberIbis2Hydrator extends Ibis2HydratorForCandidate {

self: ScheduledSpaceSubscriberPushCandidate =>

override lazy val senderId: Option[Long] = hostId

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

hostId match {

case Some(spaceHostId) =>

audioSpaceFut.map { audioSpace =>

Map(

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

"space\_id" -> spaceId,

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

}

case \_ =>

Future.exception(

new RuntimeException("Unable to get host id for ScheduledSpaceSubscriberIbis2Hydrator"))

}

}

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

mergeFutModelValues(super.modelValues, targetModelValues)

}