package com.twitter.home\_mixer.product.list\_recommended\_users.gate

import com.twitter.home\_mixer.model.request.HasListId

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.model.common.identifier.GateIdentifier

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.socialgraph.{thriftscala => sg}

import com.twitter.stitch.Stitch

import com.twitter.stitch.socialgraph.SocialGraph

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

case class ViewerIsListOwnerGate @Inject() (socialGraph: SocialGraph)

extends Gate[PipelineQuery with HasListId] {

override val identifier: GateIdentifier = GateIdentifier("ViewerIsListOwner")

private val relationship = sg.Relationship(relationshipType = sg.RelationshipType.ListOwning)

override def shouldContinue(query: PipelineQuery with HasListId): Stitch[Boolean] = {

val request = sg.ExistsRequest(

source = query.getRequiredUserId,

target = query.listId,

relationships = Seq(relationship))

socialGraph.exists(request).map(\_.exists)

}

}