package com.twitter.home\_mixer.functional\_component.gate

import com.twitter.gizmoduck.{thriftscala => t}

import com.twitter.home\_mixer.model.HomeFeatures.UserTypeFeature

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.stitch.Stitch

/\*\*

\* A Soft User is a user who is in the gradual onboarding state. This gate can be

\* used to turn off certain functionality like ads for these users.

\*/

object ExcludeSoftUserGate extends Gate[PipelineQuery] {

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

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

val softUser = query.features

.exists(\_.getOrElse(UserTypeFeature, None).exists(\_ == t.UserType.Soft))

Stitch.value(!softUser)

}

}