package com.twitter.timelineranker.observe

import com.twitter.servo.util.Gate

import com.twitter.timelineranker.model.TimelineQuery

import com.twitter.timelines.features.Features

import com.twitter.timelines.features.UserList

import com.twitter.timelines.observe.DebugObserver

import com.twitter.timelineranker.{thriftscala => thrift}

/\*\*

\* Builds the DebugObserver that is attached to thrift requests.

\* This class exists to centralize the gates that determine whether or not

\* to enable debug transcripts for a particular request.

\*/

class DebugObserverBuilder(whitelist: UserList) {

lazy val observer: DebugObserver = build()

private[this] def build(): DebugObserver = {

new DebugObserver(queryGate)

}

private[observe] def queryGate: Gate[Any] = {

val shouldEnableDebug = whitelist.userIdGate(Features.DebugTranscript)

Gate { a: Any =>

a match {

case q: thrift.EngagedTweetsQuery => shouldEnableDebug(q.userId)

case q: thrift.RecapHydrationQuery => shouldEnableDebug(q.userId)

case q: thrift.RecapQuery => shouldEnableDebug(q.userId)

case q: TimelineQuery => shouldEnableDebug(q.userId)

case \_ => false

}

}

}

}