package com.twitter.recosinjector.filters

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.recosinjector.clients.Tweetypie

import com.twitter.util.Future

class TweetFilter(

tweetypie: Tweetypie

)(

implicit statsReceiver: StatsReceiver) {

private val stats = statsReceiver.scope(this.getClass.getSimpleName)

private val requests = stats.counter("requests")

private val filtered = stats.counter("filtered")

/\*\*

\* Query Tweetypie to see if we can fetch a tweet object successfully. TweetyPie applies a safety

\* filter and will not return the tweet object if the filter does not pass.

\*/

def filterForTweetypieSafetyLevel(tweetId: Long): Future[Boolean] = {

requests.incr()

tweetypie

.getTweet(tweetId)

.map {

case Some(\_) =>

true

case \_ =>

filtered.incr()

false

}

}

}