package com.twitter.tweetypie

package hydrator

import com.twitter.spam.rtf.thriftscala.FilteredReason

import com.twitter.tweetypie.core.FilteredState

import com.twitter.tweetypie.core.ValueState

import com.twitter.tweetypie.thriftscala.\_

import com.twitter.visibility.results.counts.EngagementCounts

/\*\*

\* Redact Tweet.counts (StatusCounts) for some visibility results

\*/

object ScrubEngagementHydrator {

type Type = ValueHydrator[Option[StatusCounts], Ctx]

case class Ctx(filteredState: Option[FilteredState.Suppress])

def apply(): Type =

ValueHydrator.map[Option[StatusCounts], Ctx] { (curr: Option[StatusCounts], ctx: Ctx) =>

ctx.filteredState match {

case Some(FilteredState.Suppress(FilteredReason.SafetyResult(result))) if curr.nonEmpty =>

ValueState.delta(curr, EngagementCounts.scrubEngagementCounts(result.action, curr))

case \_ =>

ValueState.unmodified(curr)

}

}

}