package com.twitter.tweetypie

package hydrator

import com.twitter.stitch.Stitch

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.thriftscala.\_

object ReportedTweetFilter {

type Type = ValueHydrator[Unit, Ctx]

object MissingPerspectiveError

extends TweetHydrationError("Cannot determine reported state because perspective is missing")

case class Ctx(perspective: Option[StatusPerspective], underlyingTweetCtx: TweetCtx)

extends TweetCtx.Proxy

def apply(): Type =

ValueHydrator[Unit, Ctx] { (\_, ctx) =>

ctx.perspective match {

case Some(p) if !p.reported => ValueState.StitchUnmodifiedUnit

case Some(\_) => Stitch.exception(FilteredState.Unavailable.Reported)

case None => Stitch.exception(MissingPerspectiveError)

}

}.onlyIf { (\_, ctx) => ctx.opts.excludeReported && ctx.opts.forUserId.isDefined }

}