package com.twitter.tweetypie

package backends

import com.twitter.servo.exception.thriftscala

import com.twitter.servo.exception.thriftscala.ClientErrorCause

import com.twitter.stitch.Stitch

import com.twitter.storage.client.manhattan.kv.TimeoutManhattanException

import com.twitter.tweetypie.core.OverCapacity

import com.twitter.tweetypie.storage.TweetStorageClient.Ping

import com.twitter.tweetypie.storage.ClientError

import com.twitter.tweetypie.storage.RateLimited

import com.twitter.tweetypie.storage.TweetStorageClient

import com.twitter.tweetypie.util.StitchUtils

import com.twitter.util.TimeoutException

object Manhattan {

def fromClient(underlying: TweetStorageClient): TweetStorageClient =

new TweetStorageClient {

val addTweet = translateExceptions(underlying.addTweet)

val deleteAdditionalFields = translateExceptions(underlying.deleteAdditionalFields)

val getDeletedTweets = translateExceptions(underlying.getDeletedTweets)

val getTweet = translateExceptions(underlying.getTweet)

val getStoredTweet = translateExceptions(underlying.getStoredTweet)

val scrub = translateExceptions(underlying.scrub)

val softDelete = translateExceptions(underlying.softDelete)

val undelete = translateExceptions(underlying.undelete)

val updateTweet = translateExceptions(underlying.updateTweet)

val hardDeleteTweet = translateExceptions(underlying.hardDeleteTweet)

val ping: Ping = underlying.ping

val bounceDelete = translateExceptions(underlying.bounceDelete)

}

private[backends] object translateExceptions {

private[this] def pf: PartialFunction[Throwable, Throwable] = {

case e: RateLimited => OverCapacity(s"storage: ${e.getMessage}")

case e: TimeoutManhattanException => new TimeoutException(e.getMessage)

case e: ClientError => thriftscala.ClientError(ClientErrorCause.BadRequest, e.message)

}

def apply[A, B](f: A => Stitch[B]): A => Stitch[B] =

a => StitchUtils.translateExceptions(f(a), pf)

def apply[A, B, C](f: (A, B) => Stitch[C]): (A, B) => Stitch[C] =

(a, b) => StitchUtils.translateExceptions(f(a, b), pf)

}

}