package com.twitter.tweetypie.core

import com.twitter.bouncer.thriftscala.Bounce

import com.twitter.tweetypie.TweetId

import com.twitter.incentives.jiminy.thriftscala.TweetNudge

import com.twitter.tweetypie.thriftscala.PostTweetResult

import com.twitter.tweetypie.thriftscala.TweetCreateState

sealed abstract class TweetCreateFailure extends Exception {

def toPostTweetResult: PostTweetResult

}

object TweetCreateFailure {

case class Bounced(bounce: Bounce) extends TweetCreateFailure {

override def toPostTweetResult: PostTweetResult =

PostTweetResult(state = TweetCreateState.Bounce, bounce = Some(bounce))

}

case class AlreadyRetweeted(retweetId: TweetId) extends TweetCreateFailure {

override def toPostTweetResult: PostTweetResult =

PostTweetResult(state = TweetCreateState.AlreadyRetweeted)

}

case class Nudged(nudge: TweetNudge) extends TweetCreateFailure {

override def toPostTweetResult: PostTweetResult =

PostTweetResult(state = TweetCreateState.Nudge, nudge = Some(nudge))

}

case class State(state: TweetCreateState, reason: Option[String] = None)

extends TweetCreateFailure {

require(state != TweetCreateState.Bounce)

require(state != TweetCreateState.Ok)

require(state != TweetCreateState.Nudge)

override def toPostTweetResult: PostTweetResult =

PostTweetResult(state = state, failureReason = reason)

override def toString: String = s"TweetCreateFailure$$State($state, $reason)"

}

}