package com.twitter.tweetypie.util

import com.twitter.finagle.Backoff

import com.twitter.finagle.service.RetryPolicy

import com.twitter.finagle.service.RetryPolicy.RetryableWriteException

import com.twitter.servo.exception.thriftscala.ServerError

import com.twitter.util.Duration

import com.twitter.util.Throw

import com.twitter.util.TimeoutException

import com.twitter.util.Try

object RetryPolicyBuilder {

/\*\*

\* Retry on any exception.

\*/

def anyFailure[A](backoffs: Stream[Duration]): RetryPolicy[Try[A]] =

RetryPolicy.backoff[Try[A]](Backoff.fromStream(backoffs)) {

case Throw(\_) => true

}

/\*\*

\* Retry on com.twitter.util.TimeoutException

\*/

def timeouts[A](backoffs: Stream[Duration]): RetryPolicy[Try[A]] =

RetryPolicy.backoff[Try[A]](Backoff.fromStream(backoffs)) {

case Throw(\_: TimeoutException) => true

}

/\*\*

\* Retry on com.twitter.finagle.service.RetryableWriteExceptions

\*/

def writes[A](backoffs: Stream[Duration]): RetryPolicy[Try[A]] =

RetryPolicy.backoff[Try[A]](Backoff.fromStream(backoffs)) {

case Throw(RetryableWriteException(\_)) => true

}

/\*\*

\* Retry on com.twitter.servo.exception.thriftscala.ServerError

\*/

def servoServerError[A](backoffs: Stream[Duration]): RetryPolicy[Try[A]] =

RetryPolicy.backoff[Try[A]](Backoff.fromStream(backoffs)) {

case Throw(ServerError(\_)) => true

}

}