package com.twitter.tweetypie

package backends

import com.twitter.escherbird.thriftscala.TweetEntityAnnotation

import com.twitter.escherbird.{thriftscala => escherbird}

import com.twitter.finagle.service.RetryPolicy

import com.twitter.servo.util.FutureArrow

import com.twitter.tweetypie.util.RetryPolicyBuilder

object Escherbird {

import Backend.\_

type Annotate = FutureArrow[Tweet, Seq[TweetEntityAnnotation]]

def fromClient(client: escherbird.TweetEntityAnnotationService.MethodPerEndpoint): Escherbird =

new Escherbird {

val annotate = FutureArrow(client.annotate)

}

case class Config(requestTimeout: Duration, timeoutBackoffs: Stream[Duration]) {

def apply(svc: Escherbird, ctx: Backend.Context): Escherbird =

new Escherbird {

val annotate: FutureArrow[Tweet, Seq[TweetEntityAnnotation]] =

policy("annotate", requestTimeout, ctx)(svc.annotate)

}

private[this] def policy[A, B](

name: String,

requestTimeout: Duration,

ctx: Context

): Builder[A, B] =

defaultPolicy(name, requestTimeout, retryPolicy, ctx)

private[this] def retryPolicy[B]: RetryPolicy[Try[B]] =

RetryPolicyBuilder.timeouts[Any](timeoutBackoffs)

}

}

trait Escherbird {

import Escherbird.\_

val annotate: Annotate

}