package com.twitter.recos.decider

import com.twitter.decider.Decider

import com.twitter.decider.RandomRecipient

import com.twitter.util.Future

import scala.util.control.NoStackTrace

/\*

Provides deciders-controlled load shedding for a given endpoint.

The format of the decider keys is:

enable\_loadshedding\_<graphNamePrefix>\_<endpoint name>

E.g.:

enable\_loadshedding\_user-tweet-graph\_relatedTweets

Deciders are fractional, so a value of 50.00 will drop 50% of responses. If a decider key is not

defined for a particular endpoint, those requests will always be

served.

We should therefore aim to define keys for the endpoints we care most about in decider.yml,

so that we can control them during incidents.

\*/

class EndpointLoadShedder(

decider: GraphDecider) {

import EndpointLoadShedder.\_

private val keyPrefix = "enable\_loadshedding"

def apply[T](endpointName: String)(serve: => Future[T]): Future[T] = {

val key = s"${keyPrefix}\_${decider.graphNamePrefix}\_${endpointName}"

if (decider.isAvailable(key, recipient = Some(RandomRecipient)))

Future.exception(LoadSheddingException)

else serve

}

}

object EndpointLoadShedder {

object LoadSheddingException extends Exception with NoStackTrace

}