package com.twitter.home\_mixer.util

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.servo.keyvalue.KeyValueResult

import com.twitter.util.Return

import com.twitter.util.Throw

import com.twitter.util.Try

trait ObservedKeyValueResultHandler {

val statsReceiver: StatsReceiver

val statScope: String

private lazy val scopedStatsReceiver = statsReceiver.scope(statScope)

private lazy val keyTotalCounter = scopedStatsReceiver.counter("key/total")

private lazy val keyFoundCounter = scopedStatsReceiver.counter("key/found")

private lazy val keyLossCounter = scopedStatsReceiver.counter("key/loss")

private lazy val keyFailureCounter = scopedStatsReceiver.counter("key/failure")

def observedGet[K, V](

key: Option[K],

keyValueResult: KeyValueResult[K, V],

): Try[Option[V]] = {

if (key.nonEmpty) {

keyTotalCounter.incr()

keyValueResult(key.get) match {

case Return(Some(value)) =>

keyFoundCounter.incr()

Return(Some(value))

case Return(None) =>

keyLossCounter.incr()

Return(None)

case Throw(exception) =>

keyFailureCounter.incr()

Throw(exception)

case \_ =>

// never reaches here

Return(None)

}

} else {

Throw(MissingKeyException)

}

}

}