package com.twitter.servo.store

import com.twitter.finagle.stats.{StatsReceiver, Stat}

import com.twitter.servo.util.{ExceptionCounter, LogarithmicallyBucketedTimer}

import com.twitter.util.Future

class StoreObserver(statsReceiver: StatsReceiver) {

protected[this] val exceptionCounter = new ExceptionCounter(statsReceiver)

def time[T](f: => Future[T]) = {

Stat.timeFuture(statsReceiver.stat(LogarithmicallyBucketedTimer.LatencyStatName))(f)

}

def exception(ts: Throwable\*): Unit = exceptionCounter(ts)

}

class ObservableStore[K, V](underlying: Store[K, V], statsReceiver: StatsReceiver)

extends Store[K, V] {

protected[this] val observer = new StoreObserver(statsReceiver)

override def create(value: V) = observer.time {

underlying.create(value) onFailure { observer.exception(\_) }

}

override def update(value: V) = observer.time {

underlying.update(value) onFailure { observer.exception(\_) }

}

override def destroy(key: K) = observer.time {

underlying.destroy(key) onFailure { observer.exception(\_) }

}

}