package com.twitter.unified\_user\_actions.service.module

import com.google.common.cache.CacheBuilder

import com.google.inject.Provides

import com.twitter.dynmap.DynMap

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.inject.annotations.Flag

import com.twitter.unified\_user\_actions.enricher.hcache.LocalCache

import com.twitter.unified\_user\_actions.enricher.internal.thriftscala.EnrichmentKey

import com.twitter.util.Future

import java.util.concurrent.TimeUnit

import javax.inject.Singleton

object CacheModule extends TwitterModule {

private final val localCacheTtlFlagName = "local.cache.ttl.seconds"

private final val localCacheMaxSizeFlagName = "local.cache.max.size"

flag[Long](

name = localCacheTtlFlagName,

default = 1800L,

help = "Local Cache's TTL in seconds"

)

flag[Long](

name = localCacheMaxSizeFlagName,

default = 1000L,

help = "Local Cache's max size"

)

@Provides

@Singleton

def providesLocalCache(

@Flag(localCacheTtlFlagName) localCacheTtlFlag: Long,

@Flag(localCacheMaxSizeFlagName) localCacheMaxSizeFlag: Long,

statsReceiver: StatsReceiver

): LocalCache[EnrichmentKey, DynMap] = {

val underlying = CacheBuilder

.newBuilder()

.expireAfterWrite(localCacheTtlFlag, TimeUnit.SECONDS)

.maximumSize(localCacheMaxSizeFlag)

.build[EnrichmentKey, Future[DynMap]]()

new LocalCache[EnrichmentKey, DynMap](

underlying = underlying,

statsReceiver = statsReceiver.scope("enricherLocalCache"))

}

}