package com.twitter.cr\_mixer.module

import com.google.inject.Provides

import com.google.inject.Singleton

import com.twitter.app.Flag

import com.twitter.conversions.DurationOps.\_

import com.twitter.cr\_mixer.model.ModuleNames

import com.twitter.finagle.memcached.Client

import com.twitter.finagle.mtls.authentication.ServiceIdentifier

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.storehaus\_internal.memcache.MemcacheStore

import com.twitter.storehaus\_internal.util.ClientName

import com.twitter.storehaus\_internal.util.ZkEndPoint

import javax.inject.Named

object UnifiedCacheClient extends TwitterModule {

private val TIME\_OUT = 20.milliseconds

val crMixerUnifiedCacheDest: Flag[String] = flag[String](

name = "crMixer.unifiedCacheDest",

default = "/s/cache/content\_recommender\_unified\_v2",

help = "Wily path to Content Recommender unified cache"

)

val tweetRecommendationResultsCacheDest: Flag[String] = flag[String](

name = "tweetRecommendationResults.CacheDest",

default = "/s/cache/tweet\_recommendation\_results",

help = "Wily path to CrMixer getTweetRecommendations() results cache"

)

val earlybirdTweetsCacheDest: Flag[String] = flag[String](

name = "earlybirdTweets.CacheDest",

default = "/s/cache/crmixer\_earlybird\_tweets",

help = "Wily path to CrMixer Earlybird Recency Based Similarity Engine result cache"

)

@Provides

@Singleton

@Named(ModuleNames.UnifiedCache)

def provideUnifiedCacheClient(

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver,

): Client =

MemcacheStore.memcachedClient(

name = ClientName("memcache-content-recommender-unified"),

dest = ZkEndPoint(crMixerUnifiedCacheDest()),

statsReceiver = statsReceiver.scope("cache\_client"),

serviceIdentifier = serviceIdentifier,

timeout = TIME\_OUT

)

@Provides

@Singleton

@Named(ModuleNames.TweetRecommendationResultsCache)

def providesTweetRecommendationResultsCache(

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver,

): Client =

MemcacheStore.memcachedClient(

name = ClientName("memcache-tweet-recommendation-results"),

dest = ZkEndPoint(tweetRecommendationResultsCacheDest()),

statsReceiver = statsReceiver.scope("cache\_client"),

serviceIdentifier = serviceIdentifier,

timeout = TIME\_OUT

)

@Provides

@Singleton

@Named(ModuleNames.EarlybirdTweetsCache)

def providesEarlybirdTweetsCache(

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver,

): Client =

MemcacheStore.memcachedClient(

name = ClientName("memcache-crmixer-earlybird-tweets"),

dest = ZkEndPoint(earlybirdTweetsCacheDest()),

statsReceiver = statsReceiver.scope("cache\_client"),

serviceIdentifier = serviceIdentifier,

timeout = TIME\_OUT

)

}