package com.twitter.timelineranker.clients.content\_features\_cache

import com.twitter.bijection.Injection

import com.twitter.bijection.scrooge.CompactScalaCodec

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.storehaus.Store

import com.twitter.timelineranker.recap.model.ContentFeatures

import com.twitter.timelines.clients.memcache\_common.\_

import com.twitter.timelines.content\_features.{thriftscala => thrift}

import com.twitter.timelines.model.TweetId

import com.twitter.util.Duration

/\*\*

\* Content features will be stored by tweetId

\*/

class ContentFeaturesMemcacheBuilder(

config: StorehausMemcacheConfig,

ttl: Duration,

statsReceiver: StatsReceiver) {

private[this] val scalaToThriftInjection: Injection[ContentFeatures, thrift.ContentFeatures] =

Injection.build[ContentFeatures, thrift.ContentFeatures](\_.toThrift)(

ContentFeatures.tryFromThrift)

private[this] val thriftToBytesInjection: Injection[thrift.ContentFeatures, Array[Byte]] =

CompactScalaCodec(thrift.ContentFeatures)

private[this] implicit val valueInjection: Injection[ContentFeatures, Array[Byte]] =

scalaToThriftInjection.andThen(thriftToBytesInjection)

private[this] val underlyingBuilder =

new MemcacheStoreBuilder[TweetId, ContentFeatures](

config = config,

scopeName = "contentFeaturesCache",

statsReceiver = statsReceiver,

ttl = ttl

)

def build(): Store[TweetId, ContentFeatures] = underlyingBuilder.build()

}