package com.twitter.tweetypie.core

import com.twitter.servo.cache

import com.twitter.servo.cache.CachedSerializer

import com.twitter.tweetypie.thriftscala

import com.twitter.tweetypie.thriftscala.CachedTweet

import com.twitter.tweetypie.thriftscala.Tweet

import org.apache.thrift.protocol.TCompactProtocol

/\*\*

\* A container object for serializers.

\* Creates a serializer for every object type cached by the tweetypie service

\*/

object Serializer {

lazy val CompactProtocolFactory: TCompactProtocol.Factory = new TCompactProtocol.Factory

def toCached[T](underlying: cache.Serializer[T]): cache.CachedSerializer[T] =

new cache.CachedSerializer(underlying, CompactProtocolFactory)

object Tweet {

lazy val Compact: cache.ThriftSerializer[thriftscala.Tweet] =

new cache.ThriftSerializer(thriftscala.Tweet, CompactProtocolFactory)

lazy val CachedCompact: CachedSerializer[Tweet] = toCached(Compact)

}

object CachedTweet {

lazy val Compact: cache.ThriftSerializer[thriftscala.CachedTweet] =

new cache.ThriftSerializer(thriftscala.CachedTweet, CompactProtocolFactory)

lazy val CachedCompact: CachedSerializer[CachedTweet] = toCached(Compact)

}

}