package com.twitter.servo

import com.twitter.finagle.partitioning.PartitionNode

import com.twitter.servo.util.Transformer

import com.twitter.util.Try

package object cache {

type CachedValue = thriftscala.CachedValue

val CachedValue = thriftscala.CachedValue

type CachedValueStatus = thriftscala.CachedValueStatus

val CachedValueStatus = thriftscala.CachedValueStatus

type KeyTransformer[K] = K => String

type CsKeyValueResult[K, V] = KeyValueResult[K, (Try[V], Checksum)]

type KeyValueResult[K, V] = keyvalue.KeyValueResult[K, V]

val KeyValueResult = keyvalue.KeyValueResult

@deprecated("Use com.twitter.finagle.partitioning.PartitionNode instead", "1/7/2013")

type WeightedHost = PartitionNode

type Serializer[T] = Transformer[T, Array[Byte]]

/\*\*

\* Like a companion object, but for a type alias!

\*/

val Serializer = Serializers

type MemcacheFactory = (() => Memcache)

}

package cache {

package object constants {

val Colon = ":"

}

}