package com.twitter.tweetypie

package store

object Flush extends TweetStore.SyncModule {

case class Event(

tweetIds: Seq[TweetId],

flushTweets: Boolean = true,

flushCounts: Boolean = true,

logExisting: Boolean = true)

extends SyncTweetStoreEvent("flush")

trait Store {

val flush: FutureEffect[Event]

}

trait StoreWrapper extends Store { self: TweetStoreWrapper[Store] =>

override val flush: FutureEffect[Event] = wrap(underlying.flush)

}

object Store {

def apply(

cachingTweetStore: CachingTweetStore,

tweetCountsUpdatingStore: TweetCountsCacheUpdatingStore

): Store =

new Store {

override val flush: FutureEffect[Event] =

FutureEffect.inParallel(

cachingTweetStore.flush,

tweetCountsUpdatingStore.flush

)

}

}

}