package com.twitter.tweetypie

package store

import com.twitter.timelineservice.fanout.thriftscala.FanoutService

import com.twitter.tweetypie.thriftscala.\_

trait FanoutServiceStore extends TweetStoreBase[FanoutServiceStore] with AsyncInsertTweet.Store {

def wrap(w: TweetStore.Wrap): FanoutServiceStore =

new TweetStoreWrapper(w, this) with FanoutServiceStore with AsyncInsertTweet.StoreWrapper

}

object FanoutServiceStore {

val Action: AsyncWriteAction.FanoutDelivery.type = AsyncWriteAction.FanoutDelivery

def apply(

fanoutClient: FanoutService.MethodPerEndpoint,

stats: StatsReceiver

): FanoutServiceStore =

new FanoutServiceStore {

override val asyncInsertTweet: FutureEffect[AsyncInsertTweet.Event] =

FutureEffect[AsyncInsertTweet.Event] { event =>

fanoutClient.tweetCreateEvent2(

TweetCreateEvent(

tweet = event.tweet,

user = event.user,

sourceTweet = event.sourceTweet,

sourceUser = event.sourceUser,

additionalContext = event.additionalContext,

transientContext = event.transientContext

)

)

}

override val retryAsyncInsertTweet: FutureEffect[

TweetStoreRetryEvent[AsyncInsertTweet.Event]

] = TweetStore.retry(Action, asyncInsertTweet)

}

}