package com.twitter.tweetypie

package store

import com.twitter.tweetypie.thriftscala.\_

object QuotedTweetDelete extends TweetStore.SyncModule {

case class Event(

quotingTweetId: TweetId,

quotingUserId: UserId,

quotedTweetId: TweetId,

quotedUserId: UserId,

timestamp: Time,

optUser: Option[User] = None)

extends SyncTweetStoreEvent("quoted\_tweet\_delete")

with TweetStoreTweetEvent {

override def toTweetEventData: Seq[TweetEventData] =

Seq(

TweetEventData.QuotedTweetDeleteEvent(

QuotedTweetDeleteEvent(

quotingTweetId = quotingTweetId,

quotingUserId = quotingUserId,

quotedTweetId = quotedTweetId,

quotedUserId = quotedUserId

)

)

)

}

trait Store {

val quotedTweetDelete: FutureEffect[Event]

}

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

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

}

object Store {

def apply(eventBusEnqueueStore: TweetEventBusStore): Store =

new Store {

override val quotedTweetDelete: FutureEffect[Event] = eventBusEnqueueStore.quotedTweetDelete

}

}

}