package com.twitter.tweetypie

package store

import com.twitter.tseng.withholding.thriftscala.TakedownReason

import com.twitter.tweetypie.thriftscala.\_

object QuotedTweetTakedown extends TweetStore.SyncModule {

case class Event(

quotingTweetId: TweetId,

quotingUserId: UserId,

quotedTweetId: TweetId,

quotedUserId: UserId,

takedownCountryCodes: Seq[String],

takedownReasons: Seq[TakedownReason],

timestamp: Time,

optUser: Option[User] = None)

extends SyncTweetStoreEvent("quoted\_tweet\_takedown")

with TweetStoreTweetEvent {

override def toTweetEventData: Seq[TweetEventData] =

Seq(

TweetEventData.QuotedTweetTakedownEvent(

QuotedTweetTakedownEvent(

quotingTweetId = quotingTweetId,

quotingUserId = quotingUserId,

quotedTweetId = quotedTweetId,

quotedUserId = quotedUserId,

takedownCountryCodes = takedownCountryCodes,

takedownReasons = takedownReasons

)

)

)

}

trait Store {

val quotedTweetTakedown: FutureEffect[Event]

}

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

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

}

object Store {

def apply(eventBusEnqueueStore: TweetEventBusStore): Store =

new Store {

override val quotedTweetTakedown: FutureEffect[Event] =

eventBusEnqueueStore.quotedTweetTakedown

}

}

}