package com.twitter.tweetypie

package hydrator

import com.twitter.stitch.Stitch

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.repository.\_

import com.twitter.tweetypie.thriftscala.\_

/\*\*

\* Hydrates the conversationId field for any tweet that is a reply to another tweet.

\* It uses that other tweet's conversationId.

\*/

object ConversationIdHydrator {

type Type = ValueHydrator[Option[ConversationId], TweetCtx]

val hydratedField: FieldByPath =

fieldByPath(Tweet.CoreDataField, TweetCoreData.ConversationIdField)

def apply(repo: ConversationIdRepository.Type): Type =

ValueHydrator[Option[ConversationId], TweetCtx] { (\_, ctx) =>

ctx.inReplyToTweetId match {

case None =>

// Not a reply to another tweet, use tweet id as conversation root

Stitch.value(ValueState.modified(Some(ctx.tweetId)))

case Some(parentId) =>

// Lookup conversation id from in-reply-to tweet

repo(ConversationIdKey(ctx.tweetId, parentId)).liftToTry.map {

case Return(rootId) => ValueState.modified(Some(rootId))

case Throw(\_) => ValueState.partial(None, hydratedField)

}

}

}.onlyIf((curr, \_) => curr.isEmpty)

}