package com.twitter.tweetypie

package hydrator

import com.twitter.stitch.NotFound

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.repository.\_

import com.twitter.tweetypie.thriftscala.\_

object ReplyScreenNameHydrator {

import TweetLenses.Reply.{inReplyToScreenName => screenNameLens}

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

val hydratedField: FieldByPath =

fieldByPath(Tweet.CoreDataField, TweetCoreData.ReplyField, Reply.InReplyToScreenNameField)

def once(h: ValueHydrator[Option[Reply], TweetCtx]): Type =

TweetHydration.completeOnlyOnce(

hydrationType = HydrationType.ReplyScreenName,

hydrator = h

)

def apply[C](repo: UserIdentityRepository.Type): ValueHydrator[Option[Reply], C] =

ValueHydrator[Reply, C] { (reply, ctx) =>

val key = UserKey(reply.inReplyToUserId)

repo(key).liftToTry.map {

case Return(user) => ValueState.modified(screenNameLens.set(reply, Some(user.screenName)))

case Throw(NotFound) => ValueState.unmodified(reply)

case Throw(\_) => ValueState.partial(reply, hydratedField)

}

}.onlyIf((reply, \_) => screenNameLens.get(reply).isEmpty).liftOption

}