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.serverutil.DeviceSourceParser

import com.twitter.tweetypie.thriftscala.DeviceSource

import com.twitter.tweetypie.thriftscala.FieldByPath

object DeviceSourceHydrator {

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

// WebOauthId is the created\_via value for Macaw-Swift through Woodstar.

// We need to special-case it to return the same device\_source as "web",

// since we can't map multiple created\_via strings to one device\_source.

val WebOauthId: String = s"oauth:${DeviceSourceParser.Web}"

val hydratedField: FieldByPath = fieldByPath(Tweet.DeviceSourceField)

private def convertForWeb(createdVia: String) =

if (createdVia == DeviceSourceHydrator.WebOauthId) "web" else createdVia

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

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

val req = convertForWeb(ctx.createdVia)

repo(req).liftToTry.map {

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

case Throw(NotFound) => ValueState.UnmodifiedNone

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

}

}.onlyIf((curr, ctx) => curr.isEmpty && ctx.tweetFieldRequested(Tweet.DeviceSourceField))

}