package com.twitter.tweetypie

package hydrator

import com.twitter.stitch.NotFound

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.media.Media

import com.twitter.tweetypie.repository.\_

import com.twitter.tweetypie.thriftscala.\_

object MediaIsProtectedHydrator {

type Ctx = MediaEntityHydrator.Cacheable.Ctx

type Type = MediaEntityHydrator.Cacheable.Type

val hydratedField: FieldByPath = MediaEntityHydrator.hydratedField

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

ValueHydrator[MediaEntity, Ctx] { (curr, ctx) =>

val request = UserKey(ctx.userId)

repo(request).liftToTry.map {

case Return(p) => ValueState.modified(curr.copy(isProtected = Some(p)))

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

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

}

}.onlyIf { (curr, ctx) =>

// We need to update isProtected for media entities that:

// 1. Do not already have it set.

// 2. Did not come from another tweet.

//

// If the entity does not have an expandedUrl, we can't be sure

// whether the media originated with this tweet.

curr.isProtected.isEmpty &&

Media.isOwnMedia(ctx.tweetId, curr) &&

curr.expandedUrl != null

}

}