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 PlaceHydrator {

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

val HydratedField: FieldByPath = fieldByPath(Tweet.PlaceField)

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

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

val key = PlaceKey(ctx.placeId.get, ctx.opts.languageTag)

repo(key).liftToTry.map {

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

case Throw(NotFound) => ValueState.UnmodifiedNone

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

}

}.onlyIf { (curr, ctx) =>

curr.isEmpty &&

ctx.tweetFieldRequested(Tweet.PlaceField) &&

!ctx.isRetweet &&

ctx.placeId.nonEmpty

}

}