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 ContributorHydrator {

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

val hydratedField: FieldByPath = fieldByPath(Tweet.ContributorField, Contributor.ScreenNameField)

def once(h: Type): Type =

TweetHydration.completeOnlyOnce(

hydrationType = HydrationType.Contributor,

hydrator = h

)

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

ValueHydrator[Contributor, TweetCtx] { (curr, \_) =>

repo(UserKey(curr.userId)).liftToTry.map {

case Return(userIdent) => ValueState.delta(curr, update(curr, userIdent))

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

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

}

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

/\*\*

\* Updates a Contributor using the given user data.

\*/

private def update(curr: Contributor, userIdent: UserIdentity): Contributor =

curr.copy(

screenName = Some(userIdent.screenName)

)

}