package com.twitter.tweetypie

package hydrator

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.repository.\_

import com.twitter.tweetypie.thriftscala.FieldByPath

/\*\*

\* Hydrates the `conversationMuted` field of Tweet. `conversationMuted`

\* will be true if the conversation that this tweet is part of has been

\* muted by the user. This field is perspectival, so the result of this

\* hydrator should never be cached.

\*/

object ConversationMutedHydrator {

type Type = ValueHydrator[Option[Boolean], Ctx]

case class Ctx(conversationId: Option[TweetId], underlyingTweetCtx: TweetCtx)

extends TweetCtx.Proxy

val hydratedField: FieldByPath = fieldByPath(Tweet.ConversationMutedField)

private[this] val partialResult = ValueState.partial(None, hydratedField)

private[this] val modifiedTrue = ValueState.modified(Some(true))

private[this] val modifiedFalse = ValueState.modified(Some(false))

def apply(repo: ConversationMutedRepository.Type): Type = {

ValueHydrator[Option[Boolean], Ctx] { (\_, ctx) =>

(ctx.opts.forUserId, ctx.conversationId) match {

case (Some(userId), Some(convoId)) =>

repo(userId, convoId).liftToTry

.map {

case Return(true) => modifiedTrue

case Return(false) => modifiedFalse

case Throw(\_) => partialResult

}

case \_ =>

ValueState.StitchUnmodifiedNone

}

}.onlyIf { (curr, ctx) =>

// It is unlikely that this field will already be set, but if, for

// some reason, this hydrator is run on a tweet that already has

// this value set, we will skip the work to check again.

curr.isEmpty &&

// We only hydrate this field if it is explicitly requested. At

// the time of this writing, this field is only used for

// displaying UI for toggling the muted state of the relevant

// conversation.

ctx.tweetFieldRequested(Tweet.ConversationMutedField) &&

// Retweets are not part of a conversation, so should not be muted.

!ctx.isRetweet

}

}

}