package com.twitter.tweetypie.store

import com.twitter.tweetypie.Tweet

import com.twitter.tweetypie.serverutil.ExtendedTweetMetadataBuilder

import com.twitter.tweetypie.thriftscala.EditControl

import com.twitter.tweetypie.thriftscala.InitialTweetUpdateRequest

import com.twitter.tweetypie.util.EditControlUtil

/\* Logic to update the initial tweet with new information when that tweet is edited \*/

object InitialTweetUpdate {

/\* Given the initial tweet and update request, copy updated edit

\* related fields onto it.

\*/

def updateTweet(initialTweet: Tweet, request: InitialTweetUpdateRequest): Tweet = {

// compute a new edit control initial with updated list of edit tweet ids

val editControl: EditControl.Initial =

EditControlUtil.editControlForInitialTweet(initialTweet, request.editTweetId).get()

// compute the correct extended metadata for a permalink

val extendedTweetMetadata =

request.selfPermalink.map(link => ExtendedTweetMetadataBuilder(initialTweet, link))

initialTweet.copy(

selfPermalink = initialTweet.selfPermalink.orElse(request.selfPermalink),

editControl = Some(editControl),

extendedTweetMetadata = initialTweet.extendedTweetMetadata.orElse(extendedTweetMetadata)

)

}

}