package com.twitter.tweetypie

package hydrator

import com.twitter.tweetypie.thriftscala.\_

/\*\*

\* Removes superfluous urls entities when there is a corresponding MediaEntity for the same

\* url.

\*/

object SuperfluousUrlEntityScrubber {

case class RawEntity(fromIndex: Short, toIndex: Short, url: String)

object RawEntity {

def from(e: UrlEntity): RawEntity = RawEntity(e.fromIndex, e.toIndex, e.url)

def fromUrls(es: Seq[UrlEntity]): Set[RawEntity] = es.map(from(\_)).toSet

def from(e: MediaEntity): RawEntity = RawEntity(e.fromIndex, e.toIndex, e.url)

def fromMedia(es: Seq[MediaEntity]): Set[RawEntity] = es.map(from(\_)).toSet

}

val mutation: Mutation[Tweet] =

Mutation[Tweet] { tweet =>

val mediaEntities = getMedia(tweet)

val urlEntities = getUrls(tweet)

if (mediaEntities.isEmpty || urlEntities.isEmpty) {

None

} else {

val mediaUrls = mediaEntities.map(RawEntity.from(\_)).toSet

val scrubbedUrls = urlEntities.filterNot(e => mediaUrls.contains(RawEntity.from(e)))

if (scrubbedUrls.size == urlEntities.size)

None

else

Some(TweetLenses.urls.set(tweet, scrubbedUrls))

}

}

}