package com.twitter.tweetypie

package repository

import com.twitter.stitch.SeqGroup

import com.twitter.stitch.Stitch

import com.twitter.tweetypie.media.MediaMetadata

import com.twitter.tweetypie.media.MediaMetadataRequest

object MediaMetadataRepository {

type Type = MediaMetadataRequest => Stitch[MediaMetadata]

def apply(getMediaMetadata: FutureArrow[MediaMetadataRequest, MediaMetadata]): Type = {

// use an `SeqGroup` to group the future-calls together, even though they can be

// executed independently, in order to help keep hydration between different tweets

// in sync, to improve batching in hydrators which occur later in the pipeline.

val requestGroup = SeqGroup[MediaMetadataRequest, MediaMetadata] {

requests: Seq[MediaMetadataRequest] =>

Future.collect(requests.map(r => getMediaMetadata(r).liftToTry))

}

mediaReq => Stitch.call(mediaReq, requestGroup)

}

}