package com.twitter.tweetypie

package repository

import com.twitter.stitch.Stitch

import com.twitter.stitch.compat.LegacySeqGroup

import com.twitter.tweetypie.backends.Escherbird

import com.twitter.tweetypie.thriftscala.EscherbirdEntityAnnotations

object EscherbirdAnnotationRepository {

type Type = Tweet => Stitch[Option[EscherbirdEntityAnnotations]]

def apply(annotate: Escherbird.Annotate): Type =

// use a `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.

tweet =>

Stitch

.call(tweet, LegacySeqGroup(annotate.liftSeq))

.map { annotations =>

if (annotations.isEmpty) None

else Some(EscherbirdEntityAnnotations(annotations))

}

}