package com.twitter.tweetypie.thriftscala

import com.twitter.util.Future

/\*\*

\* A trait for TweetService implementations that wrap an underlying

\* TweetService and need to modify only some of the methods.

\*/

trait TweetServiceProxy extends TweetService.MethodPerEndpoint {

protected def underlying: TweetService.MethodPerEndpoint

/\*\*

\* Default implementation simply passes through the Future but logic can be added to wrap each

\* invocation to the underlying TweetService

\*/

protected def wrap[A](f: => Future[A]): Future[A] =

f

override def getTweets(request: GetTweetsRequest): Future[Seq[GetTweetResult]] =

wrap(underlying.getTweets(request))

override def getTweetFields(request: GetTweetFieldsRequest): Future[Seq[GetTweetFieldsResult]] =

wrap(underlying.getTweetFields(request))

override def getTweetCounts(request: GetTweetCountsRequest): Future[Seq[GetTweetCountsResult]] =

wrap(underlying.getTweetCounts(request))

override def setAdditionalFields(request: SetAdditionalFieldsRequest): Future[Unit] =

wrap(underlying.setAdditionalFields(request))

override def deleteAdditionalFields(request: DeleteAdditionalFieldsRequest): Future[Unit] =

wrap(underlying.deleteAdditionalFields(request))

override def postTweet(request: PostTweetRequest): Future[PostTweetResult] =

wrap(underlying.postTweet(request))

override def postRetweet(request: RetweetRequest): Future[PostTweetResult] =

wrap(underlying.postRetweet(request))

override def unretweet(request: UnretweetRequest): Future[UnretweetResult] =

wrap(underlying.unretweet(request))

override def getDeletedTweets(

request: GetDeletedTweetsRequest

): Future[Seq[GetDeletedTweetResult]] =

wrap(underlying.getDeletedTweets(request))

override def deleteTweets(request: DeleteTweetsRequest): Future[Seq[DeleteTweetResult]] =

wrap(underlying.deleteTweets(request))

override def updatePossiblySensitiveTweet(

request: UpdatePossiblySensitiveTweetRequest

): Future[Unit] =

wrap(underlying.updatePossiblySensitiveTweet(request))

override def undeleteTweet(request: UndeleteTweetRequest): Future[UndeleteTweetResponse] =

wrap(underlying.undeleteTweet(request))

override def eraseUserTweets(request: EraseUserTweetsRequest): Future[Unit] =

wrap(underlying.eraseUserTweets(request))

override def incrTweetFavCount(request: IncrTweetFavCountRequest): Future[Unit] =

wrap(underlying.incrTweetFavCount(request))

override def deleteLocationData(request: DeleteLocationDataRequest): Future[Unit] =

wrap(underlying.deleteLocationData(request))

override def scrubGeo(request: GeoScrub): Future[Unit] =

wrap(underlying.scrubGeo(request))

override def takedown(request: TakedownRequest): Future[Unit] =

wrap(underlying.takedown(request))

override def flush(request: FlushRequest): Future[Unit] =

wrap(underlying.flush(request))

override def incrTweetBookmarkCount(request: IncrTweetBookmarkCountRequest): Future[Unit] =

wrap(underlying.incrTweetBookmarkCount(request))

}