package com.twitter.search.earlybird.common;

import org.apache.thrift.TException;

import org.apache.thrift.TSerializer;

import org.apache.thrift.protocol.TSimpleJSONProtocol;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.search.earlybird.thrift.EarlybirdRequest;

import com.twitter.search.earlybird.thrift.EarlybirdResponse;

public class RequestResponseForLogging {

private static final Logger LOG = LoggerFactory.getLogger(

RequestResponseForLogging.class);

private static final Logger FAILED\_REQUEST\_LOG = LoggerFactory.getLogger(

RequestResponseForLogging.class.getName() + ".FailedRequests");

private final EarlybirdRequest request;

private final EarlybirdResponse response;

public RequestResponseForLogging(EarlybirdRequest request,

EarlybirdResponse response) {

this.request = request;

this.response = response;

}

private String serialize(EarlybirdRequest clearedRequest, EarlybirdResponse theResponse) {

TSerializer serializer = new TSerializer(new TSimpleJSONProtocol.Factory());

try {

String requestJson = serializer.toString(clearedRequest);

String responseJson = serializer.toString(theResponse);

return "{\"request\":" + requestJson + ", \"response\":" + responseJson + "}";

} catch (TException e) {

LOG.error("Failed to serialize request/response for logging.", e);

return "";

}

}

/\*\*

\* Logs the request and response stored in this instance to the failure log file.

\*/

public void logFailedRequest() {

// Do the serializing/concatting this way so it happens on the background thread for

// async logging

FAILED\_REQUEST\_LOG.info("{}", new Object() {

@Override

public String toString() {

return serialize(

EarlybirdRequestUtil.copyAndClearUnnecessaryValuesForLogging(request), response);

}

});

}

}