package com.twitter.search.earlybird\_root;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

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

public class EarlybirdProtectedValidationBehavior extends EarlybirdServiceValidationBehavior {

private static final Logger LOG =

LoggerFactory.getLogger(EarlybirdProtectedValidationBehavior.class);

@Override

public EarlybirdResponse getResponseIfInvalidRequest(EarlybirdRequest request) {

if (!request.isSetSearchQuery() || request.getSearchQuery() == null) {

String errorMsg = "Invalid EarlybirdRequest, no ThriftSearchQuery specified. " + request;

LOG.warn(errorMsg);

return createErrorResponse(errorMsg);

}

ThriftSearchQuery searchQuery = request.getSearchQuery();

// Make sure this request is valid for the protected tweets cluster.

if (!searchQuery.isSetFromUserIDFilter64() || searchQuery.getFromUserIDFilter64().isEmpty()) {

String errorMsg = "ThriftSearchQuery.fromUserIDFilter64 not set. " + request;

LOG.warn(errorMsg);

return createErrorResponse(errorMsg);

}

if (!searchQuery.isSetSearcherId()) {

String errorMsg = "ThriftSearchQuery.searcherId not set. " + request;

LOG.warn(errorMsg);

return createErrorResponse(errorMsg);

}

if (searchQuery.getSearcherId() < 0) {

String errorMsg = "Invalid ThriftSearchQuery.searcherId: " + searchQuery.getSearcherId()

+ ". " + request;

LOG.warn(errorMsg);

return createErrorResponse(errorMsg);

}

return super.getResponseIfInvalidRequest(request);

}

}