package com.twitter.search.earlybird\_root.validators;

import com.twitter.search.common.schema.earlybird.EarlybirdCluster;

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

import com.twitter.util.Future;

public class SearchResultsValidator

implements ServiceResponseValidator<EarlybirdResponse> {

private final EarlybirdCluster cluster;

public SearchResultsValidator(EarlybirdCluster cluster) {

this.cluster = cluster;

}

@Override

public Future<EarlybirdResponse> validate(EarlybirdResponse response) {

if (!response.isSetSearchResults()

|| !response.getSearchResults().isSetResults()) {

return Future.exception(

new IllegalStateException(cluster + " didn't set search results"));

} else if (!response.getSearchResults().isSetMaxSearchedStatusID()) {

return Future.exception(

new IllegalStateException(cluster + " didn't set max searched status id"));

} else {

boolean isEarlyTerminated = response.isSetEarlyTerminationInfo()

&& response.getEarlyTerminationInfo().isEarlyTerminated();

if (!isEarlyTerminated && !response.getSearchResults().isSetMinSearchedStatusID()) {

return Future.exception(

new IllegalStateException(

cluster + " neither early terminated nor set min searched status id"));

} else {

return Future.value(response);

}

}

}

}