package com.twitter.search.ingester.pipeline.strato\_fetchers;

import java.util.List;

import java.util.Set;

import java.util.stream.Collectors;

import com.twitter.periscope.api.thriftjava.AudioSpacesLookupContext;

import com.twitter.stitch.Stitch;

import com.twitter.strato.catalog.Fetch;

import com.twitter.strato.client.Client;

import com.twitter.strato.client.Fetcher;

import com.twitter.strato.data.Conv;

import com.twitter.strato.thrift.TBaseConv;

import com.twitter.ubs.thriftjava.AudioSpace;

import com.twitter.util.Future;

import com.twitter.util.Try;

/\*\*

\* Fetches from the audio space core strato column.

\*/

public class AudioSpaceCoreFetcher {

private static final String CORE\_STRATO\_COLUMN = "";

private static final AudioSpacesLookupContext

EMPTY\_AUDIO\_LOOKUP\_CONTEXT = new AudioSpacesLookupContext();

private final Fetcher<String, AudioSpacesLookupContext, AudioSpace> fetcher;

public AudioSpaceCoreFetcher(Client stratoClient) {

fetcher = stratoClient.fetcher(

CORE\_STRATO\_COLUMN,

true, // enables checking types against catalog

Conv.stringConv(),

TBaseConv.forClass(AudioSpacesLookupContext.class),

TBaseConv.forClass(AudioSpace.class));

}

public Future<Fetch.Result<AudioSpace>> fetch(String spaceId) {

return Stitch.run(fetcher.fetch(spaceId, EMPTY\_AUDIO\_LOOKUP\_CONTEXT));

}

/\*\*

\* Use stitch to fetch mulitiple AudioSpace Objects at once

\*/

public Future<List<Try<Fetch.Result<AudioSpace>>>> fetchBulkSpaces(Set<String> spaceIds) {

return Stitch.run(

Stitch.collectToTry(

spaceIds

.stream()

.map(spaceId -> fetcher.fetch(spaceId, EMPTY\_AUDIO\_LOOKUP\_CONTEXT))

.collect(Collectors.toList())

)

);

}

}