package com.twitter.search.earlybird.segment;

import java.io.IOException;

import java.util.Optional;

import com.twitter.search.common.indexing.thriftjava.ThriftVersionedEvents;

import com.twitter.search.common.util.io.recordreader.RecordReader;

import com.twitter.search.earlybird.document.TweetDocument;

import com.twitter.search.earlybird.partition.SegmentInfo;

/\*\*

\* SegmentDataReaderSet provides an interface to create and manage the various

\* RecordReaders used to index Earlybird segments.

\*/

public interface SegmentDataReaderSet {

/\*\*

\* Instruct the document RecordReaders (i.e. document, geo, ... as appropriate) to read from this

\* segment.

\*/

void attachDocumentReaders(SegmentInfo segmentInfo) throws IOException;

/\*\*

\* Instruct the reader set to add segment to non-document RecordReaders (deletes, features, etc.)

\*/

void attachUpdateReaders(SegmentInfo segmentInfo) throws IOException;

/\*\*

\* Mark a segment as "complete", denoting that we are done reading document records from it.

\*

\* This instructs the reader set to stop reading documents from the segment (if it hasn't

\* already), although for now geo-document records can still be read. Updates RecordReaders

\* (deletes, etc.) may continue to read entries for the segment.

\*/

void completeSegmentDocs(SegmentInfo segmentInfo);

/\*\*

\* This instructs the reader set to stop reading updates for the Segment. It

\* should remove the segment from all non-document RecordReaders (deletes, etc.)

\*/

void stopSegmentUpdates(SegmentInfo segmentInfo);

/\*\*

\* Stops all RecordReaders and closes all resources.

\*/

void stopAll();

/\*\*

\* Returns true if all RecordReaders are 'caught up' with the data sources they

\* are reading from. This might mean that the end of a file has been reached,

\* or that we are waiting/polling for new records from an append-only database.

\*/

boolean allCaughtUp();

/\*\*

\* Create a new DocumentReader for the given segment that is not managed by this set.

\*/

RecordReader<TweetDocument> newDocumentReader(SegmentInfo segmentInfo) throws Exception;

/\*\*

\* Returns the document reader for the current segment.

\*/

RecordReader<TweetDocument> getDocumentReader();

/\*\*

\* Returns a combined update events reader for all segments.

\*/

RecordReader<ThriftVersionedEvents> getUpdateEventsReader();

/\*\*

\* Returns the update events reader for the given segment.

\*/

RecordReader<ThriftVersionedEvents> getUpdateEventsReaderForSegment(SegmentInfo segmentInfo);

/\*\*

\* Returns the offset in the update events stream for the given segment that this earlybird should

\* start indexing from.

\*/

Optional<Long> getUpdateEventsStreamOffsetForSegment(SegmentInfo segmentInfo);

}