package com.twitter.search.ingester.pipeline.twitter;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import com.google.common.collect.Lists;

import com.twitter.common\_internal.text.version.PenguinVersion;

import com.twitter.search.common.debug.thriftjava.DebugEvents;

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

import com.twitter.search.common.schema.thriftjava.ThriftDocument;

import com.twitter.search.common.schema.thriftjava.ThriftField;

import com.twitter.search.common.schema.thriftjava.ThriftFieldData;

import com.twitter.search.common.schema.thriftjava.ThriftIndexingEvent;

import com.twitter.search.common.schema.thriftjava.ThriftIndexingEventType;

import com.twitter.search.ingester.model.IngesterThriftVersionedEvents;

/\*\*

\* Converter for {@code ThriftVersionedEvents}.

\*

\*/

public class ThriftVersionedEventsConverter {

private static final long UNUSED\_USER\_ID = -1L;

private Iterable<PenguinVersion> penguinVersions;

public ThriftVersionedEventsConverter(Iterable<PenguinVersion> penguinVersions) {

this.penguinVersions = penguinVersions;

}

/\*\*

\* Creates a DELETE IngesterThriftVersionedEvents instance for the given tweet ID and user ID.

\*

\* @param tweetId The tweet ID.

\* @param userId The user ID.

\* @param debugEvents The DebugEvents to propagate to the returned IngesterThriftVersionedEvents

\* instance.

\* @return A DELETE IngesterThriftVersionedEvents instance with the given tweet and user IDs.

\*/

public IngesterThriftVersionedEvents toDelete(

long tweetId, long userId, DebugEvents debugEvents) {

ThriftIndexingEvent thriftIndexingEvent = new ThriftIndexingEvent()

.setEventType(ThriftIndexingEventType.DELETE)

.setUid(tweetId);

return toThriftVersionedEvents(tweetId, userId, thriftIndexingEvent, debugEvents);

}

/\*\*

\* Creates an OUT\_OF\_ORDER\_APPEND IngesterThriftVersionedEvents instance for the given tweet ID

\* and the given value for the given field.

\*

\* @param tweetId The tweet ID.

\* @param field The updated field.

\* @param value The new field value.

\* @param debugEvents The DebugEvents to propagate to the returned IngesterThriftVersionedEvents

\* instance.

\* @return An OUT\_OF\_ORDER\_APPEND IngesterThriftVersionedEvents instance with the given tweet ID

\* and value for the field.

\*/

public IngesterThriftVersionedEvents toOutOfOrderAppend(

long tweetId,

EarlybirdFieldConstants.EarlybirdFieldConstant field,

long value,

DebugEvents debugEvents) {

ThriftField updateField = new ThriftField()

.setFieldConfigId(field.getFieldId())

.setFieldData(new ThriftFieldData().setLongValue(value));

ThriftDocument document = new ThriftDocument()

.setFields(Lists.newArrayList(updateField));

ThriftIndexingEvent thriftIndexingEvent = new ThriftIndexingEvent()

.setEventType(ThriftIndexingEventType.OUT\_OF\_ORDER\_APPEND)

.setUid(tweetId)

.setDocument(document);

return toThriftVersionedEvents(tweetId, UNUSED\_USER\_ID, thriftIndexingEvent, debugEvents);

}

/\*\*

\* Creates a PARTIAL\_UPDATE IngesterThriftVersionedEvents instance for the given tweet ID and the

\* given value for the given feature.

\*

\* @param tweetId The tweet ID.

\* @param feature The updated feature.

\* @param value The new feature value.

\* @param debugEvents The DebugEvents to propagate to the returned IngesterThriftVersionedEvents

\* instance.

\* @return A PARTIAL\_UPDATE IngesterThriftVersionedEvents instance with the given tweet ID and

\* value for the feature.

\*/

public IngesterThriftVersionedEvents toPartialUpdate(

long tweetId,

EarlybirdFieldConstants.EarlybirdFieldConstant feature,

int value,

DebugEvents debugEvents) {

ThriftField updateField = new ThriftField()

.setFieldConfigId(feature.getFieldId())

.setFieldData(new ThriftFieldData().setIntValue(value));

ThriftDocument document = new ThriftDocument()

.setFields(Lists.newArrayList(updateField));

ThriftIndexingEvent thriftIndexingEvent = new ThriftIndexingEvent()

.setEventType(ThriftIndexingEventType.PARTIAL\_UPDATE)

.setUid(tweetId)

.setDocument(document);

return toThriftVersionedEvents(tweetId, UNUSED\_USER\_ID, thriftIndexingEvent, debugEvents);

}

// Wraps the given ThriftIndexingEvent into a ThriftVersionedEvents instance.

private IngesterThriftVersionedEvents toThriftVersionedEvents(

long tweetId, long userId, ThriftIndexingEvent thriftIndexingEvent, DebugEvents debugEvents) {

if (!thriftIndexingEvent.isSetCreateTimeMillis()

&& (debugEvents != null)

&& debugEvents.isSetCreatedAt()) {

thriftIndexingEvent.setCreateTimeMillis(debugEvents.getCreatedAt().getEventTimestampMillis());

}

Map<Byte, ThriftIndexingEvent> versionedEvents = new HashMap<>();

for (PenguinVersion penguinVersion : penguinVersions) {

versionedEvents.put(penguinVersion.getByteValue(), thriftIndexingEvent);

}

IngesterThriftVersionedEvents events =

new IngesterThriftVersionedEvents(userId, versionedEvents);

events.setId(tweetId);

events.setDebugEvents(debugEvents);

return events;

}

public void updatePenguinVersions(List<PenguinVersion> updatePenguinVersions) {

penguinVersions = updatePenguinVersions;

}

}