package com.twitter.search.ingester.model;

import java.util.Map;

import com.google.common.primitives.Longs;

import com.twitter.search.common.debug.DebugEventAccumulator;

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

import com.twitter.search.common.partitioning.base.Partitionable;

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

/\*\*

\* Wrap of ThriftVersionedEvents, make it partitionable for the queue writer.

\*/

public class IngesterThriftVersionedEvents extends ThriftVersionedEvents

implements Comparable<ThriftVersionedEvents>, Partitionable, DebugEventAccumulator {

// Make userId field easier to be accessed to calculate partition number

private final long userId;

public IngesterThriftVersionedEvents(long userId) {

this.userId = userId;

}

public IngesterThriftVersionedEvents(long userId,

Map<Byte, ThriftIndexingEvent> versionedEvents) {

super(versionedEvents);

this.userId = userId;

}

public IngesterThriftVersionedEvents(long userId, ThriftVersionedEvents original) {

super(original);

this.userId = userId;

}

@Override

public int compareTo(ThriftVersionedEvents o) {

return Longs.compare(getId(), o.getId());

}

@Override

public long getTweetId() {

return this.getId();

}

@Override

public long getUserId() {

return this.userId;

}

}