package com.twitter.search.ingester.model;

import java.util.List;

import javax.annotation.Nullable;

import com.google.common.base.Preconditions;

import com.google.common.primitives.Longs;

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

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

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

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

import com.twitter.search.common.relevance.entities.TwitterMessage;

/\*\*

\* A Twitter "status" object (e.g. a message)

\*

\*/

public class IngesterTwitterMessage extends TwitterMessage

implements Comparable<IndexerStatus>, IndexerStatus, Partitionable {

private final DebugEvents debugEvents;

public IngesterTwitterMessage(Long twitterId, List<PenguinVersion> supportedPenguinVersions) {

this(twitterId, supportedPenguinVersions, null);

}

public IngesterTwitterMessage(

Long twitterId,

List<PenguinVersion> penguinVersions,

@Nullable DebugEvents debugEvents) {

super(twitterId, penguinVersions);

this.debugEvents = debugEvents == null ? new DebugEvents() : debugEvents.deepCopy();

}

@Override

public int compareTo(IndexerStatus o) {

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

}

@Override

public boolean equals(Object o) {

return (o instanceof IngesterTwitterMessage)

&& compareTo((IngesterTwitterMessage) o) == 0;

}

@Override

public int hashCode() {

return HashPartitionFunction.hashCode(getId());

}

public boolean isIndexable(boolean indexProtectedTweets) {

return getFromUserScreenName().isPresent()

&& getId() != INT\_FIELD\_NOT\_PRESENT

&& (indexProtectedTweets || !isUserProtected());

}

@Override

public long getTweetId() {

return this.getId();

}

@Override

public long getUserId() {

Preconditions.checkState(getFromUserTwitterId().isPresent(), "The author user ID is missing");

return getFromUserTwitterId().get();

}

@Override

public DebugEvents getDebugEvents() {

return debugEvents;

}

}