package com.twitter.search.common.schema.earlybird;

import com.google.common.base.Preconditions;

import com.twitter.search.common.encoding.features.IntegerEncodedFeatures;

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

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

import com.twitter.search.common.schema.SchemaUtil;

import com.twitter.search.common.schema.base.FeatureConfiguration;

import com.twitter.search.common.schema.base.ImmutableSchemaInterface;

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

/\*\*

\* A class for encoding earlybird features in integers

\*/

public abstract class EarlybirdEncodedFeatures extends IntegerEncodedFeatures {

private final ImmutableSchemaInterface schema;

private final EarlybirdFieldConstant baseField;

public EarlybirdEncodedFeatures(ImmutableSchemaInterface schema,

EarlybirdFieldConstant baseField) {

this.schema = schema;

this.baseField = baseField;

}

/\*\*

\* Write this object into packedFeatures of the given VersionedTweetFeatures.

\*/

public void writeFeaturesToVersionedTweetFeatures(

VersionedTweetFeatures versionedTweetFeatures) {

if (!versionedTweetFeatures.isSetPackedFeatures()) {

versionedTweetFeatures.setPackedFeatures(new PackedFeatures());

}

copyToPackedFeatures(versionedTweetFeatures.getPackedFeatures());

}

/\*\*

\* Write this object into extendedPackedFeatures of the given VersionedTweetFeatures.

\*/

public void writeExtendedFeaturesToVersionedTweetFeatures(

VersionedTweetFeatures versionedTweetFeatures) {

if (!versionedTweetFeatures.isSetExtendedPackedFeatures()) {

versionedTweetFeatures.setExtendedPackedFeatures(new PackedFeatures());

}

copyToPackedFeatures(versionedTweetFeatures.getExtendedPackedFeatures());

}

@Override

public String toString() {

StringBuilder ret = new StringBuilder();

ret.append("Tweet features: \n");

for (FeatureConfiguration feature

: EarlybirdSchemaCreateTool.FEATURE\_CONFIGURATION\_MAP.values()) {

ret.append(feature.getName()).append(": ").append(getFeatureValue(feature)).append("\n");

}

return ret.toString();

}

public boolean isFlagSet(EarlybirdFieldConstant field) {

return isFlagSet(schema.getFeatureConfigurationById(field.getFieldId()));

}

public int getFeatureValue(EarlybirdFieldConstant field) {

return getFeatureValue(schema.getFeatureConfigurationById(field.getFieldId()));

}

public EarlybirdEncodedFeatures setFlag(EarlybirdFieldConstant field) {

setFlag(schema.getFeatureConfigurationById(field.getFieldId()));

return this;

}

public EarlybirdEncodedFeatures clearFlag(EarlybirdFieldConstant field) {

clearFlag(schema.getFeatureConfigurationById(field.getFieldId()));

return this;

}

public EarlybirdEncodedFeatures setFlagValue(EarlybirdFieldConstant field,

boolean value) {

setFlagValue(schema.getFeatureConfigurationById(field.getFieldId()), value);

return this;

}

public EarlybirdEncodedFeatures setFeatureValue(EarlybirdFieldConstant field,

int value) {

setFeatureValue(schema.getFeatureConfigurationById(field.getFieldId()), value);

return this;

}

public EarlybirdEncodedFeatures setFeatureValueIfGreater(EarlybirdFieldConstant field,

int value) {

setFeatureValueIfGreater(schema.getFeatureConfigurationById(field.getFieldId()), value);

return this;

}

public boolean incrementIfNotMaximum(EarlybirdFieldConstant field) {

return incrementIfNotMaximum(schema.getFeatureConfigurationById(field.getFieldId()));

}

private static final class ArrayEncodedTweetFeatures extends EarlybirdEncodedFeatures {

private final int[] encodedInts;

private ArrayEncodedTweetFeatures(ImmutableSchemaInterface schema,

EarlybirdFieldConstant baseField) {

super(schema, baseField);

final int numIntegers = SchemaUtil.getCSFFieldFixedLength(schema, baseField.getFieldId());

Preconditions.checkState(numIntegers > 0);

this.encodedInts = new int[numIntegers];

}

@Override

public int getNumInts() {

return encodedInts.length;

}

@Override

public int getInt(int pos) {

return encodedInts[pos];

}

@Override

public void setInt(int pos, int value) {

encodedInts[pos] = value;

}

}

/\*\*

\* Create a new {@link EarlybirdEncodedFeatures} object based on schema and base field.

\* @param schema the schema for all fields

\* @param baseField base field's constant value

\*/

public static EarlybirdEncodedFeatures newEncodedTweetFeatures(

ImmutableSchemaInterface schema, EarlybirdFieldConstant baseField) {

return new ArrayEncodedTweetFeatures(schema, baseField);

}

/\*\*

\* Create a new {@link EarlybirdEncodedFeatures} object based on schema and base field name.

\* @param schema the schema for all fields

\* @param baseFieldName base field's name

\*/

public static EarlybirdEncodedFeatures newEncodedTweetFeatures(

ImmutableSchemaInterface schema, String baseFieldName) {

EarlybirdFieldConstant baseField = EarlybirdFieldConstants.getFieldConstant(baseFieldName);

Preconditions.checkNotNull(baseField);

return newEncodedTweetFeatures(schema, baseField);

}

}