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

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

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

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

/\*\*

\* Utility APIs for ThriftDocument.

\*/

public final class ThriftDocumentUtil {

private ThriftDocumentUtil() {

}

/\*\*

\* Get ThriftField out of a ThriftDocument.

\*/

public static ThriftField getField(ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

int id = idMap.getFieldID(fieldName);

for (ThriftField field : thriftDoc.getFields()) {

int fieldId = field.getFieldConfigId();

if (fieldId == id) {

return field;

}

}

return null;

}

/\*\*

\* Get all fields out of a ThriftDocument that match the given field name.

\*/

public static List<ThriftField> getFields(

ThriftDocument thriftDoc, String fieldName, FieldNameToIdMapping idMap) {

int id = idMap.getFieldID(fieldName);

List<ThriftField> result = new ArrayList<>();

for (ThriftField field : thriftDoc.getFields()) {

int fieldId = field.getFieldConfigId();

if (fieldId == id) {

result.add(field);

}

}

return result;

}

/\*\*

\* Retrieve the long value from a thrift field

\*/

public static long getLongValue(ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

ThriftField f = getField(thriftDoc, fieldName, idMap);

return f == null ? 0L : f.getFieldData().getLongValue();

}

/\*\*

\* Retrieve the byte value from a thrift field

\*/

public static byte getByteValue(ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

ThriftField f = getField(thriftDoc, fieldName, idMap);

return f == null ? (byte) 0 : f.getFieldData().getByteValue();

}

/\*\*

\* Retrieve the bytes value from a thrift field

\*/

public static byte[] getBytesValue(ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

ThriftField f = getField(thriftDoc, fieldName, idMap);

return f == null ? null : f.getFieldData().getBytesValue();

}

/\*\*

\* Retrieve the int value from a thrift field

\*/

public static int getIntValue(ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

ThriftField f = getField(thriftDoc, fieldName, idMap);

return f == null ? 0 : f.getFieldData().getIntValue();

}

/\*\*

\* Retrieve the string value from a thrift field

\*/

public static String getStringValue(ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

ThriftField f = getField(thriftDoc, fieldName, idMap);

return f == null ? null : f.getFieldData().getStringValue();

}

/\*\*

\* Retrieve the string values from all thrift fields with the given fieldName.

\*/

public static List<String> getStringValues(

ThriftDocument thriftDoc,

String fieldName,

FieldNameToIdMapping idMap) {

List<ThriftField> fields = getFields(thriftDoc, fieldName, idMap);

List<String> fieldStrings = new ArrayList<>();

for (ThriftField field : fields) {

fieldStrings.add(field.getFieldData().getStringValue());

}

return fieldStrings;

}

/\*\*

\* Returns whether the specified document has duplicate fields.

\*/

public static boolean hasDuplicateFields(ThriftDocument thriftDoc) {

Set<Integer> seen = new HashSet<>();

for (ThriftField field : thriftDoc.getFields()) {

if (!seen.add(field.getFieldConfigId())) {

return true;

}

}

return false;

}

/\*\*

\* Get ThriftField out of a ThriftDocument.

\*/

public static ThriftField getField(ThriftDocument thriftDoc, int fieldId) {

for (ThriftField field : thriftDoc.getFields()) {

if (field.getFieldConfigId() == fieldId) {

return field;

}

}

return null;

}

}