**Convertir CSV a JSON**

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.SerializationFeature;

import com.fasterxml.jackson.databind.node.ArrayNode;

import com.fasterxml.jackson.databind.node.ObjectNode;

import java.io.\*;

import java.nio.charset.StandardCharsets;

public class CsvToJsonConverter {

public static void main(String[] args) {

String csvFile = "ruta\_del\_archivo.csv"; // Reemplaza con la ruta de tu archivo CSV

String line;

String[] headers = null;

**ObjectMapper** objectMapper = new ObjectMapper();

objectMapper.enable(SerializationFeature.INDENT\_OUTPUT); // Habilitar la sangría del JSON

**ArrayNode** jsonArray = objectMapper.createArrayNode();

try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(csvFile), StandardCharsets.UTF\_8))) {

// Leer la primera línea para obtener los encabezados

if ((line = br.readLine()) != null) {

**headers = line.split(";");**

}

while ((line = br.readLine()) != null) {

String[] values = line.split(";");

**ObjectNode**  jsonObject = objectMapper.createObjectNode();

for (int i = 0; i < headers.length; i++) {

jsonObject.put(headers[i], values[i]);

}

jsonArray.add(jsonObject);

}

} catch (IOException e) {

e.printStackTrace();

}

// Convertir el JSONArray a una cadena JSON formateada con sangría y codificada en UTF-8

try {

String jsonString = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(jsonArray);

System.out.println(jsonString);

} catch (IOException e) {

e.printStackTrace();

}

}

}

# Clase ObjectNode

La clase ObjectNode es una parte de la biblioteca Jackson, que es ampliamente utilizada en Java para trabajar con JSON. Jackson proporciona una serie de clases para representar y manipular datos JSON en Java de manera eficiente. La clase ObjectNode es una de esas clases y se utiliza para representar objetos JSON en Java.

En términos simples, un objeto JSON es una colección de pares clave-valor donde las claves son cadenas y los valores pueden ser de varios tipos, incluyendo otros objetos JSON, números, cadenas, arreglos, booleanos, o null. La clase ObjectNode se utiliza para representar un objeto JSON en Java y permite agregar, modificar y acceder a los valores y propiedades dentro de ese objeto.

Algunas de las operaciones comunes que puedes realizar con ObjectNode incluyen:

* Agregar propiedades: Puedes agregar pares clave-valor al objeto JSON usando el **método put(key, value).**
* Obtener propiedades: Puedes obtener el valor asociado con una clave específica usando el **método get(key).**
* Eliminar propiedades: Puedes eliminar una propiedad específica usando el **método remove(key).**
* Verificar si una propiedad existe: Puedes verificar si una propiedad específica existe en el objeto JSON usando el **método has(key).**
* Iterar a través de las propiedades: Puedes iterar a través de todas las propiedades del objeto JSON utilizando un bucle.

Ejemplo simple de cómo usar ObjectNode para crear un objeto JSON:

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.node.ObjectNode;

public class ObjectNodeExample {

public static void main(String[] args) {

ObjectMapper objectMapper = new ObjectMapper();

ObjectNode jsonObject = objectMapper.createObjectNode();

// Agregar propiedades al objeto JSON

jsonObject.put("nombre", "John");

jsonObject.put("edad", 30);

jsonObject.put("soltero", true);

// Convertir el objeto JSON a una cadena JSON

String jsonString = jsonObject.toString();

System.out.println(jsonString);

}

}

En este ejemplo, primero creamos un objeto ObjectNode vacío, luego agregamos propiedades al objeto usando el método put, y finalmente convertimos el objeto JSON a una cadena JSON utilizando toString(). Esto genera la cadena JSON: {"nombre":"John","edad":30,"soltero":true}.

En tu código anterior, utilizaste ObjectNode para construir objetos JSON a partir de los datos del archivo CSV y luego convertiste esos objetos JSON en una cadena JSON formateada con sangría utilizando Jackson para que sean más legibles.

# ArrayNode

La clase ArrayNode también es parte de la biblioteca Jackson y se utiliza para representar arreglos JSON en Java. Al igual que ObjectNode, ArrayNode es una clase que facilita la manipulación y construcción de arreglos JSON en un programa Java.

Un arreglo JSON es una secuencia ordenada de valores, donde cada valor puede ser de cualquier tipo JSON válido, incluyendo otros objetos JSON, arreglos, números, cadenas, booleanos o null. ArrayNode se utiliza para representar esta estructura de datos en Java.

Algunas de las operaciones comunes que puedes realizar con ArrayNode incluyen:

* Agregar elementos: Puedes agregar elementos al arreglo usando el método **add(value).**
* Obtener elementos: Puedes obtener un elemento específico del arreglo usando el método **get(index).**
* Iterar a través de elementos: Puedes recorrer todos los elementos del arreglo utilizando un bucle.
* Obtener el tamaño del arreglo: Puedes obtener el número de elementos en el arreglo usando el método **size().**
* Convertir el arreglo a una lista Java: Puedes convertir el ArrayNode a una lista Java utilizando **toList().**

Ejemplo simple de cómo usar ArrayNode para crear un arreglo JSON:

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.node.ArrayNode;

public class ArrayNodeExample {

public static void main(String[] args) {

ObjectMapper objectMapper = new ObjectMapper();

**ArrayNode** jsonArray = objectMapper.createArrayNode();

// Agregar elementos al arreglo

jsonArray.add("Manzana");

jsonArray.add("Banana");

jsonArray.add("Cereza");

// Convertir el arreglo a una cadena JSON

String jsonString = jsonArray.toString();

System.out.println(jsonString);

}

}

En este ejemplo, primero creamos un objeto ArrayNode vacío, luego agregamos elementos al arreglo usando el método add, y finalmente convertimos el arreglo JSON en una cadena JSON utilizando toString(). Esto genera la cadena JSON: ["Manzana","Banana","Cereza"].

En tu código anterior, utilizaste ArrayNode para construir arreglos JSON para representar múltiples registros (filas) del archivo CSV. Cada elemento del ArrayNode representaba un registro en el JSON final.