GSON

GSON es una biblioteca de Google que permite convertir objetos Java a **JSON** y viceversa de manera sencilla.

Instalación de GSON

Si usas **Maven**, agrega esta dependencia en el `pom.xml`:

```
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.10.1</version>
</dependency>
```

Si usas **Gradle**, agrégalo en `build.gradle`:

```
    implementation 'com.google.code.gson:gson:2.10.1'
```

---

1. Convertir un Objeto Java a JSON (Serialización)

```
import com.google.gson.Gson;
class Persona {
    String nombre;
    int edad;
    public Persona(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
}
public class ConvertirAJson {
    public static void main(String[] args) {
    Persona persona = new Persona("Juan", 25); // Convertir objeto a JSON
    Gson gson = new Gson();
    String json = gson.toJson(persona);
    System.out.println("JSON generado: " + json);
    }
}
```

Salida:

```
{"nombre":"Juan","edad":25}
```

---

2. Convertir JSON a Objeto Java (Deserialización)

```
import com.google.gson.Gson;

public class ConvertirDesdeJson {
    public static void main(String[] args) {
        String json = "{\"nombre\":\"Ana\",\"edad\":30}";  // Convertir JSON
a objeto Java
        Gson gson = new Gson();
        Persona persona = gson.fromJson(json, Persona.class);
        System.out.println("Nombre: " + persona.nombre);
        System.out.println("Edad: " + persona.edad);
    }
}
```

Salida:

```
Nombre: Ana Edad: 30
```

---

3. Trabajar con Listas de Objetos

```
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import java.lang.reflect.Type;
import java.util.Arrays;
import java.util.List;

public class ListaJson {
    public static void main(String[] args) {
        List<Persona> personas = Arrays.asList( new Persona("Carlos", 28),new
Persona("Lucía", 24));
        Gson gson = new Gson();
        String json = gson.toJson(personas);
        System.out.println("JSON Lista: " + json); // Convertir JSON a lista
de objetos
        Type tipLista = new TypeToken<List<Persona>>() {}.getType();
        List<Persona> listaPersonas = gson.fromJson(json, tipoLista);
```

```
        for (Persona p : listaPersonas) {
            System.out.println("Nombre: " + p.nombre + ", Edad: " + p.edad);
        }
    }
}
```

---

4. Ignorar Campos con `@Expose`

Si solo quieres incluir ciertos campos en la serialización, usa `@Expose` junto con `GsonBuilder`.

```
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.annotations.Expose;

class Usuario {
    @Expose
    String usuario;
    @Expose
    String email;
    transient String password; // No se serializa

    public Usuario(String usuario, String email, String password) {
        this.usuario = usuario;
        this.email = email;
        this.password = password;
    }
}
public class IgnorarCampos {
    public static void main(String[] args) {
        Usuario user = new Usuario("pepe123", "pepe@mail.com", "secreto123");
        Gson gson = new
GsonBuilder().excludeFieldsWithoutExposeAnnotation().create();
        String json = gson.toJson(user);
        System.out.println("JSON: " + json);
    }
}
```

Salida:

```
{"usuario":"pepe123","email":"pepe@mail.com"}
```