



# GSON Introducción

## [1. Introducción](#)

## [2. Gson: Convertir objetos Java a JSON y viceversa](#)

## [3. Características de Gson](#)

## [4. Configuración y descarga](#)

## [5. Prerrequisitos](#)

## [6. Paquetes y Clases de Gson](#)

## [Resumen:](#)

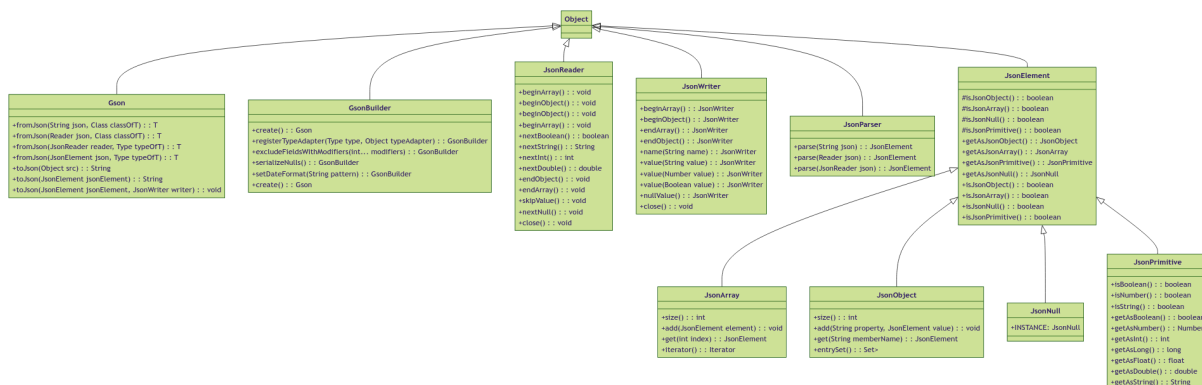
## 1. Introducción

- **Gson** es una biblioteca de Google para Java que permite convertir objetos Java a JSON y viceversa.
- Originalmente creado para uso interno en Google, luego se liberó como proyecto de código abierto.
- Su propósito principal es **facilitar la serialización** (de Java a JSON) y **deserialización** (de JSON a Java) **de objetos**.

### Componentes principales de Gson:

1. **Gson**: Analiza objetos JSON a objetos Java y viceversa (métodos `fromJson()` y `toJson()`).
2. **JsonReader**: Analiza flujos de datos JSON, token por token.

3. **JsonParser** : Convierte JSON a una estructura de árbol de objetos Java específicos de Gson.



!!!!Imagen Super importante!!!! Expandir para ver mejor

## 2. Gson: Convertir objetos Java a JSON y viceversa

- **Funcionalidad principal:** Convertir objetos Java en su representación JSON y convertir cadenas JSON a objetos Java equivalentes.
- **Ventajas de Gson:**
  - No requiere anotaciones en las clases Java, lo que permite trabajar con objetos sin acceso a su código fuente.
  - Soporta genéricos de Java.
  - Permite la conversión de objetos arbitrariamente complejos, incluidos aquellos con jerarquías de herencia profundas.

## 3. Características de Gson

- **Métodos principales:**
  - **toJson()** : Convierte objetos Java a JSON.
  - **fromJson()** : Convierte JSON a objetos Java.
- **Soporte para objetos preexistentes:** Gson maneja objetos de los que no se tiene el código fuente.

- **Compatibilidad con genéricos de Java:** Permite la conversión de tipos complejos.
  - **Representación personalizada:** Permite crear representaciones personalizadas de objetos.
- 

## 4. Configuración y descarga

Dependiendo del tipo de proyecto que se esté utilizando, existen diferentes maneras de incluir Gson:

- **Con Gradle:**

```
dependencies {  
    implementation 'com.google.code.gson:gson:2.11.0'  
}
```

- **Con Maven:**

```
<dependency>  
    <groupId>com.google.code.gson</groupId>  
    <artifactId>gson</artifactId>  
    <version>2.11.0</version>  
</dependency>
```

- **Descarga del archivo JAR de Gson:**

Si no usas Maven, puedes descargar el archivo JAR desde [Maven Central](#).

- El archivo JAR más reciente es `gson-2.10.1.jar`.
- **Proceso de descarga:** Haz clic en el enlace de descargas, selecciona "jar", y guarda el archivo.
- Luego, se debe incluir en el `CLASSPATH` al compilar y ejecutar el código:

```
javac -cp gson-2.10.1.jar archivo_fuente  
java -cp gson-2.10.1.jar;. archivo_clase_principal
```

---

## 5. Prerrequisitos

- **Versión mínima de Java SE:**
    - Gson 2.9.0 y posterior requiere **Java 7**.
    - Gson 2.8.9 y versiones anteriores requieren **Java 6**.
  - **Dependencias de JPMS (Java 9+):**
    - **java.sql** (opcional desde Gson 2.8.9): Para adaptadores predeterminados de clases SQL.
    - **jdk.unsigned** (opcional): Para usar la clase `Unsafe` en entornos que lo permitan (para crear instancias sin constructor por defecto).
  - **Nivel mínimo de API de Android:**
    - Gson 2.11.0 y posterior requiere **API nivel 21**.
    - Gson 2.10.1 y anteriores pueden funcionar con **API nivel 19**.
- 

## 6. Paquetes y Clases de Gson

Gson está compuesto por varios paquetes que incluyen más de 30 clases e interfaces. Los principales paquetes son:

1. **com.google.gson:** Contiene la clase principal **Gson** para trabajar con JSON.
  2. **com.google.gson.annotations:** Proporciona tipos de anotaciones para ser utilizados con Gson.
  3. **com.google.gson.reflect:** Contiene clases para obtener información de tipo de objetos genéricos.
  4. **com.google.gson.stream:** Proporciona clases de utilidad para leer y escribir JSON.
- 

## Resumen:

Gson es una biblioteca poderosa y flexible para manejar la conversión entre objetos Java y JSON sin necesidad de realizar cambios en el código fuente original de los objetos Java. Soporta genéricos y tipos complejos, lo que la hace ideal para trabajar con datos JSON en proyectos Java y Android. Se

puede integrar fácilmente en proyectos mediante Maven, Gradle o descargando el archivo JAR directamente.