

JSon

- JavaScript Object Notation.
- Formato de texto sencillo para el intercambio de datos.
- Es más sencillo escribir un parser para JSON que para XML.

JSon

```
{  
  "title": "Nuevos cursos en DigitalHouse",  
  "date": "19/10/2016",  
  "description": "Hola",  
  "author": "andresF"  
}
```

GSon

- Es una librería que nos permite trabajar de forma sencilla con JSON.
- Facilita el serializado y deserializado de objetos.

Paso a paso

1) Como toda librería, debemos importarla, agregando la dependencia en el Gradle.

```
dependencies {  
    ...  
    compile 'com.google.code.gson:gson:2.3.1'  
    ...  
}
```

Paso a paso

2) Implementar el modelo del objeto (POJO) que queremos parsear con JSON.

Recordar:

- Los atributos que debemos parsear deben tener el mismo nombre con el que vienen en el Json. Sino, se deberá especificar explícitamente que lo modificaremos.

En el caso de colecciones de datos, se deberá crear:

- Una clase que modele a cada elemento de la colección.
- Una clase cuyo atributo sea dicha colección.

Paso a paso

3) Implementar un método en el DAO que será el encargado de leer el Json y crear el objeto POJO.

```
//Creamos un stream para leer el archivo Json.
```

```
AssetManager manager = context.getAssets();
```

```
InputStream newsJson = manager.open("news.json");
```

```
BufferedReader bufferedReaderIn = new BufferedReader(new InputStreamReader(newsJson));
```

```
//Creamos un Objeto de la clase Gson que me permitirá sencillamente parsear el Json.
```

```
Gson gson = new Gson();
```

```
//Utilizando el objeto gson y el método fromJson, realizamos el parsing el archivo que tenemos en el  
bufferedReaderIn y utilizando como "molde" la clase News.
```

```
News news = gson.fromJson(bufferedReaderIn, News.class);
```