

# PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

## ANEXO TEMA 9:

## CREAR API REST CON JSON SERVER



# ÍNDICE

1. ¿Qué es JSON Server?
2. Instalar Node.js
3. Instalar JSON Server
4. Crear archivo JSON (db.json)
5. Arrancar el servidor y desplegar la API Rest
6. Configurar la BASE\_URL para el emulador



# 1.- ¿Qué es JSON Server?

- JSON Server es una herramienta de Node.js que permite crear un servidor local simulando una API Rest con operaciones CRUD, utilizando un archivo JSON como fuente de datos.

## 2.- Instalar Node.js

- En primer lugar tenemos que instalar Node.js en nuestro sistema operativo. Podemos descargarlo desde el siguiente enlace:

<https://nodejs.org/en/download>

## 3.- Instalar JSON Server

- Con Node.js instalado, posteriormente debemos instalar la herramienta JSON Server en el sistema operativo. Para ello, abrimos el terminal del sistema y ejecutamos el siguiente comando:

```
npm install -g json-server
```

- Con la opción “-g” nos aseguramos de que el paquete se instale globalmente en todo el sistema y así poder usarlo en todos nuestros proyectos.

## 4.- Crear archivo JSON (db.json)

- A continuación crearemos el archivo JSON que utilizará nuestra API Rest como base de datos, con el nombre **db.json**.
- Ejemplo:

```
db.json
1 {
2   "cars": [
3     {
4       "id": 1,
5       "brand": "Ford",
6       "model": "Focus",
7       "year": 2017
8     },
9
10    {
11      "id": 2,
12      "brand": "Renault",
13      "model": "Megane",
14      "year": 2021
15    },
16
17    {
18      "id": 1,
19      "brand": "Volkswagen",
20      "model": "Golf",
21      "year": 2010
22    },
23
24    {
25      "id": 1,
26      "brand": "Audi",
27      "model": "A4",
28      "year": 2019
29    }
30  ]
31 }
```

## 5.- Arrancar el servidor y desplegar la API Rest

- Para arrancar el servidor y desplegar la API Rest debemos abrir un terminal del sistema, acceder a la ruta donde se encuentra el archivo *db.json* que hemos creado previamente y ejecutar el siguiente comando:

```
json-server --watch db.json
```

- Es importante arrancar el servidor en modo observación, con la opción "**--watch**", para que analice los cambios que se producen en el archivo JSON y actualice la API Rest cuando proceda.

## 5.- Arrancar el servidor y desplegar la API Rest

- Si el servidor se ha iniciado correctamente y la API se ha desplegado, podremos ver lo siguiente en el terminal:

```
\{^_^}/ hi!
```

```
Loading db.json  
Done
```

```
Resources  
http://localhost:3000/cars
```

```
Home  
http://localhost:3000
```

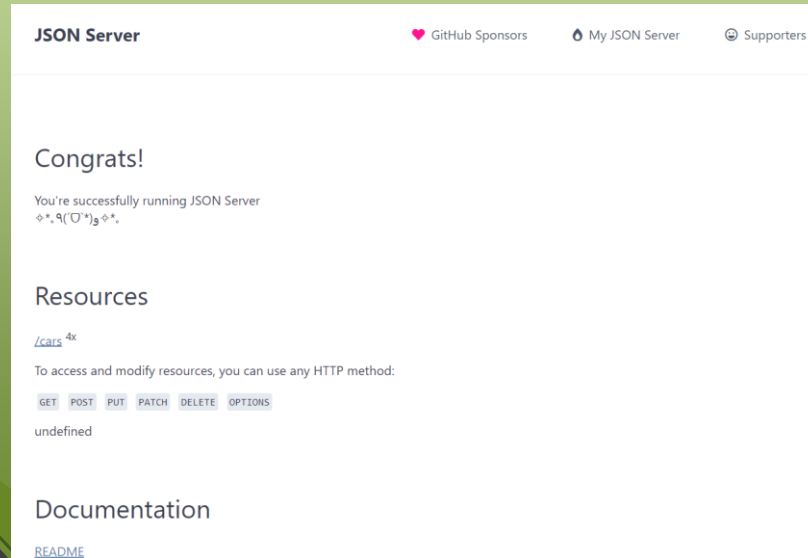
```
Type s + enter at any time to create a snapshot of the database  
Watching...
```



## 5.- Arrancar el servidor y desplegar la API Rest

- <http://localhost:3000/>  
Home de la API Rest

- <http://localhost:3000/cars>  
Recursos de la base de datos de nuestra API Rest



```
[
  {
    "id": 1,
    "brand": "Ford",
    "model": "Focus",
    "year": 2017
  },
  {
    "id": 2,
    "brand": "Renault",
    "model": "Megane",
    "year": 2021
  },
  {
    "id": 1,
    "brand": "Volkswagen",
    "model": "Golf",
    "year": 2010
  },
  {
    "id": 1,
    "brand": "Audi",
    "model": "A4",
    "year": 2019
  }
]
```

## 5.- Arrancar el servidor y desplegar la API Rest

- El servidor JSON crea automáticamente los siguientes endpoints para nuestra API Rest:
  - GET /cars
  - GET /cars/{id}
  - POST /cars
  - PUT /cars/{id}
  - PATCH /cars/{id}
  - DELETE /cars/{id}

## 5.- Arrancar el servidor y desplegar la API Rest

- Si queremos acceder a nuestra API desde un cliente externo (por ejemplo, desde el emulador de Android Studio) debemos arrancar el servidor en el host **0.0.0.0**, indicándole también el puerto:

```
json-server --host 0.0.0.0 --port 3000 --watch db.json
```

## 6. Configurar la BASE\_URL para el emulador

- Para que podamos acceder a nuestra API Rest desde el emulador de Android Studio debemos configurar la BASE\_URL que usa el objeto Retrofit, cambiando la IP local (localhost o 127.0.0.1) por el alias correspondiente a esta IP del router virtual con el que se conecta el emulador (10.0.2.2):

```
private val BASE_URL = "http://10.0.2.2:3000/"
```

- En el siguiente enlace podemos ver cómo configurar las redes para el emulador de Android Studio:

<https://developer.android.com/studio/run/emulator-networking?hl=es-419>