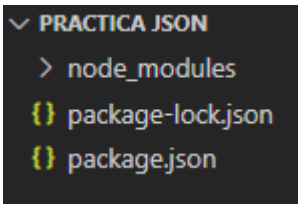
Creació d'un Fake Api Rest

Per poder crear-la, primer hem d'instal·lar Node.js:

Un cop instal·lat, ja es pot treballar amb Visual Studio Code amb línia de comandes. Amb npm init -y creem els packages a la carpeta que vulguem crear l'API:

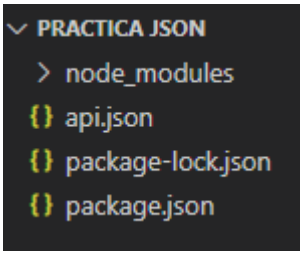
```
PS C:\Users\Oscar\Documents\M04\Practica JSON> npm init -y
Wrote to C:\Users\Oscar\Documents\M04\Practica JSON\package.json:

{
  "name": "practica-json",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "servicio": "json-server --watch api.json"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "json-server": "^0.17.2"
  },
  "dependencies": {
    "accepts": "^1.3.8",
    "acorn": "^8.8.2",
    "acorn-jsx": "^5.3.2",
    "ajv": "^6.12.6",
    "ansi-regex": "^5.0.1",
    "ansi-styles": "^4.3.0",
    "argparse": "^2.0.1",
    "array-flatten": "^1.1.1",
    "array-includes": "^3.1.6",
    "array.prototype.flat": "^1.3.1",
    "array.prototype.flatmap": "^1.3.1",
    "array.prototype.tosorted": "^1.1.1",
    "available-typed-arrays": "^1.0.5",
    "balanced-match": "^1.0.2",
```



Instal·lem la dependència json-server, que es copiaràn a la carpeta node_modules i s'afegiràn a les dependències del desenvolupador del package:

```
PS C:\Users\Oscar\Documents\M04\Practica JSON> npm i json-server -D

up to date, audited 312 packages in 2s

88 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Oscar\Documents\M04\Practica JSON>
```

Creem l'arxiu de dades amb extensió .json a la carpeta on tenim els packages:

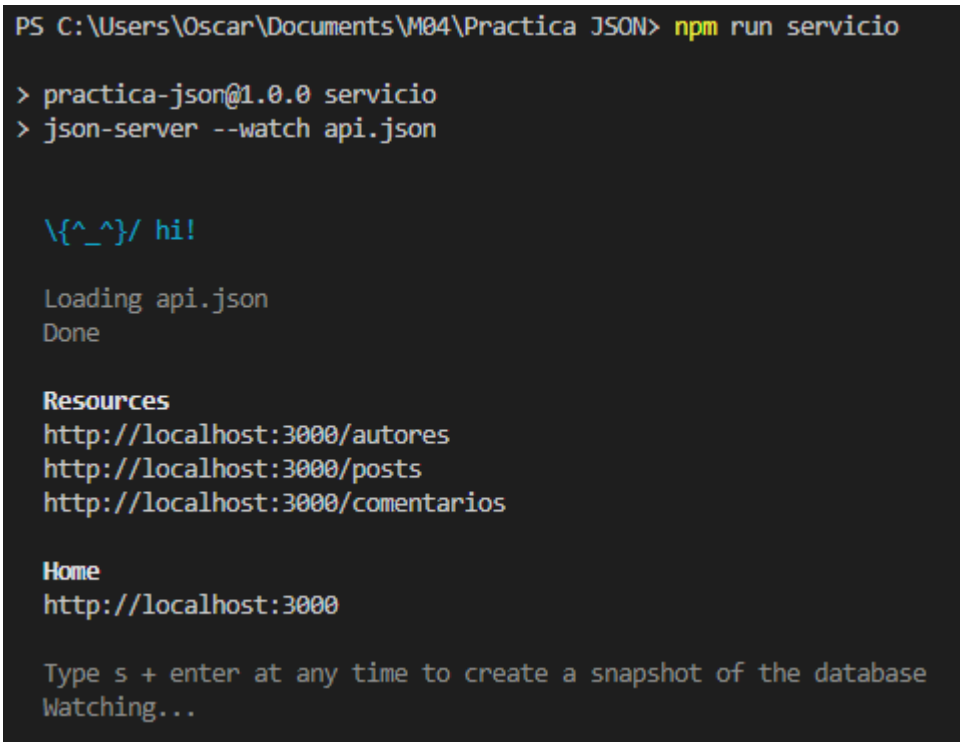


Afegim objectes amb tantes propietats com branques que vulguem que tingui l'API:

```json
{
  "autores": [
    { "id": 1, "nombre": "José Manuel Alarcón" },
    { "id": 2, "nombre": "campusMVP"}
  ],
  "posts": [
    { "id": 1, "titulo": "Cómo crear una API falsa en 1 minuto", "autor": 1},
    { "id": 2, "titulo": "Aprender a programar con campusMVP", "autor": 2}
  ],
  "comentarios": [
    { "id": 1, "mensaje": "¡Los mejores cursos que he hecho!", "postId": 2},
    { "id": 2, "mensaje": "Un curso genial!", "postId": 2}
  ]
}
```

Creem un script per utilitzar la dependència json-server instal·lada prèviament al package i executar-lo amb l'arxiu de dades que hem creat:

L'executem des de la línia de comandes indicant el nom de la comanda del script creat:

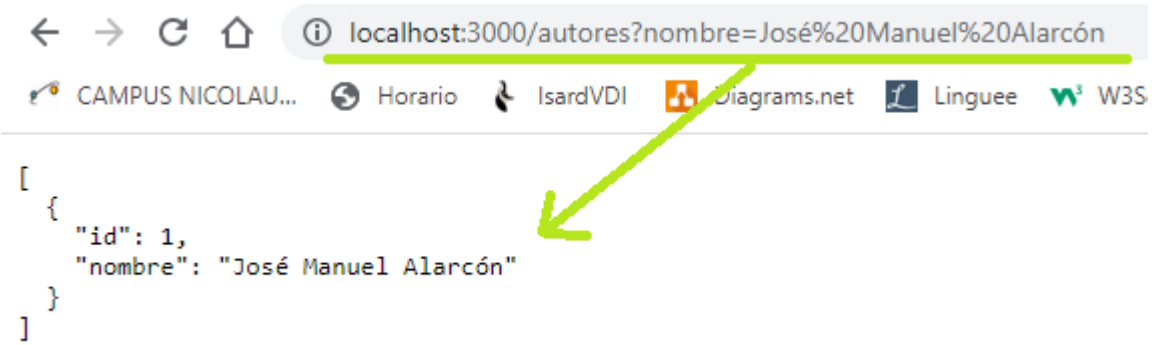


De manera automàtica utilitza el port 3000 per cada recurs. Això es pot canviar afegint "--port" i indicant-li el número de port que volem utilitzar:

```
"scripts": {
  "servicio": "json-server --port 8080 --watch api.json"
```

```
PS C:\Users\Oscar\Documents\M04\Practica JSON> npm run servicio

> practica-json@1.0.0 servicio
> json-server --port 8080 --watch api.json


  \{^_^}/ hi!

  Loading api.json
  Done

  Resources
  http://localhost:8080/autores
  http://localhost:8080/posts
  http://localhost:8080/comentarios

  Home
  http://localhost:8080

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

Una vegada iniciat, es poden llegir les dades en un navegador (Ctrl+click al recurs) i fer consultes en la mateixa barra de cerca:



O també es poden escriure dades, per exemple a través de https://hoppscotch.io/:

1 - Canviar el mètode GET a POST.
2 - Introduir l'enllaç corresponent al recurs al que volem afegir dades.
3 - Seleccionar el tipus d'introducció per cos, en comptes de per paràmetres.
4 - Indicar el tipus de contingut que s'afegirà (en aquest cas, un comentari) i l'indentificador del post al que s'afegirà. També es pot indicar l'identificador del missatge, però en cas de no fer-ho, s'hi assignarà automàticament el següent identificador lliure.
5 - Clic en el botó "Enviar".

Si s'han introduït correctament les dades apareixerà l'estat "**201 · Created**" en verd:

Es pot comprovar que s'ha realitzat correctament l'operació si mirem el contingut del nostre fitxer api.json, ja que en la secció "comentarios" apareix el nou comentari amb id 3.


```
{
  "autores": [
    { "id": 1, "nombre": "José Manuel Alarcón"},
    { "id": 2, "nombre": "campusMVP"}
  ],
  "posts": [
    { "id": 1, "titulo": "Cómo crear una API falsa en 1 minuto", "autor": 1},
    { "id": 2, "titulo": "Aprender a programar con campusMVP", "autor": 2}
  ],
  "comentarios": [
    { "id": 1, "mensaje": "¡Los mejores cursos que he hecho!", "postId": 2},
    { "id": 2, "mensaje": "Un curso genial!", "postId": 2},
    { "mensaje": "Nuevo comentario", "postId": 1, "id": 3}
  ]
}
```