

APUNTES NODE.JS - Abel Rios - 13/05/2022

Repaso de cómo crear un servidor:

- Crear una Carpeta y abrirla en node (VisualStudio)
- npm init → License: MIT
- Creamos index.js (dentro de la carpeta)
- npm install express
- npm install nodemon
- package.json → En scripts añadimos: "dev": "nodemon index.js"
- Para levantar el servidor → npm run dev

En la parte superior del index vamos a poner:

```
const express = require("express");
const app = express();
app.use(express.json());
```

y en la parte inferior:

```
const PORT = 8080;
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

Creando un endpoint:

app.verbo("url", ...

↓

GET → obtener

PUT → actualizar

POST → crear

DELETE → borrar

}
}

Lo elegiremos según lo que queramos hacer

Esqueleto del endpoint:

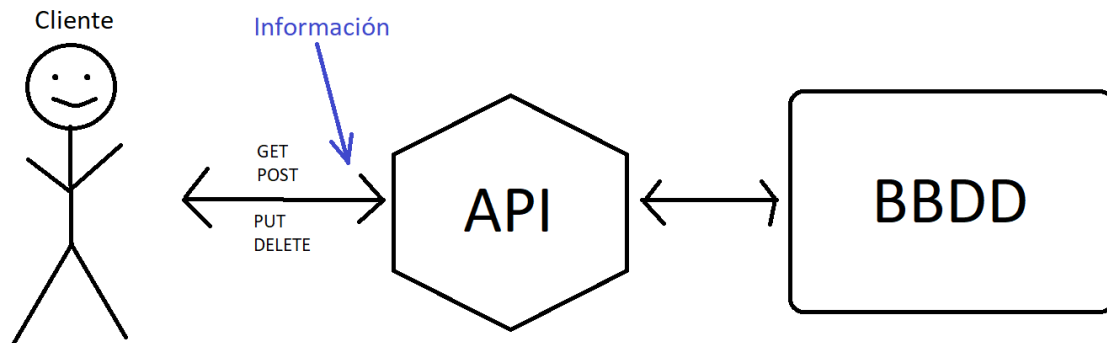
```
app.get("/", (request, response) => {
  // ...
});
```

→ Toda la información la contiene "request".

→ La respuesta la devuelve response.json(), y es la información que la API va a devolver.

Estructura:

(microservices.io → arquitectura de API's)



Cliente manda información a la API (Get, Post, Put, Delete):

- XML
- JSON
- Multipart

API procesa información:

- Mapping de Objetos
- Acceso a BBDD (crear, eliminar...)
- Tratamiento de información

BBDD (consistencia de datos):

- Relacional (MySQL)
- No Relacional (MongoDB)

Formas de Envío de Datos

QueryString → `api/usuario?nombre=abel&apellido=rios`
→ Los parámetros empiezan después de la “?”
→ Se anidan usando “&”

Params → `api/usuario/abel/rios` → Barra de navegación / Postman
→ `api/usuario/:nombre/:apellido` → URL servidor

Body → Postman / Aplicación Cliente
→ Se manda la información generalmente como XML o JSON
→ Se usa para mandar **Post** o **Put**

*** La **barra del navegador** SÓLO entiende el Get , en el Postman sí podemos hacer Delete a través de URL ***

¿Qué es Postman?

Software que simula la respuesta de un cliente (lo que manda la app al servidor) y obtiene la respuesta del servidor.

Sólo y únicamente manda información al servidor.

FIN DEL RESUMEN

(Ejercicios en el archivo index de la carpeta Server2)

Ejercicio 1:

Crear:

- Un array de 3 objetos "Notas"
- Notas :
 - id (int)
 - from (string)
 - text (string)
 - for (string)

2 endpoints que devuelva una nota (querystring y params)

- QueryString: <http://localhost:3001/notaquery?id=1>
- Params: <http://localhost:3001/notaparams/1> (donde el número final es el id de la nota)

1 endpoint para crear una nota (body/postman)

- Postman: ejerciciosNode / crearnota
****OJO**** en el body tiene que ir la opción "raw" y en formato JSON

2 endpoint para eliminar una nota (postman)

- Postman: ejerciciosNode / borrarNotaQuery
- Postman: ejerciciosNode/ borrarNotaParams

1 endpoint que intercambie el texto de dos notas:

- Postman: ejerciciosNode / intercambiarTexto (usando params)

Ejercicio 2:

const alumnos =

```
{
  dni:"00984564Y",
  telefono:"654786907",
  nombre:"Adrian",
  apellido:"Molina",
  edad:21,
  sexo:"Masculino"
},
{
  dni:"9874521J",
  telefono:"645789214",
  nombre:"Javier",
  apellido:"Gutierrez",
  edad:23,
  sexo:"Masculino"
},
{
  dni:"85632149O",
  telefono:"632459871",
  nombre:"Miguel",
  apellido:"Torres",
  edad:23,
  sexo:"Masculino"
},
{
  dni:"68542175U",
  telefono:"654786907",
  nombre:"Omar",
  apellido:"Montes",
  edad:24,
  sexo:"Masculino"
},
{
  dni:"58478962L",
  telefono:"685632641",
  nombre:"Paula",
  apellido:"Garcia",
  edad:24,
  sexo:"Femenino"
},
{
  dni:"6897452U",
  telefono:"653214789",
  nombre:"Marta",
```

```
    apellido:"Gonzalez",
    edad:24,
    sexo:"Femenino"
  },
  {
    dni:"68741298H",
    telefono:"693654217",
    nombre:"Marta",
    apellido:"Gutierrez",
    edad:23,
    sexo:"Femenino"
  }
]
```

1 - Sacar todos los alumnos que tienen 23 años

- <http://localhost:3001/seleccionarporedad/23>

2 - Sacar aquellas personas que tienen el sexo femenino y no tienen 23 años

- <http://localhost:3001/seleccionarporsexoynoedad/Femenino/23>

3 - Introducir un alumno nuevo, si su DNI ya existe, entonces debe de sacar un mensaje de error y no crear el alumno

- Postman: ejerciciosNode / introducirAlumno

4 - Introducir un nuevo alumno, si ese alumno ya existe en la lista (DNI ya existe) , entonces actualizar sus datos, si no, crearlo de nuevo.

- Postman: ejerciciosNode / introducirAlumnoActualizar

5 - Eliminar los alumnos que tengan menos 23 años (23 incluido)

- Postman: ejerciciosNode / eliminarAlumnos

6 - Eliminar aquellos alumnos cuyo nombre empiece por M

- Postman: ejerciciosNode / eliminarnombre