



Hola Bienvenido!

Registra tu asistencia

<https://forms.gle/6hfdT5U5v5fF34RQ8>



Introducción al uso de APIs con Node.js y Express

Objetivo general

Comprender qué es una API, cómo se comunican las aplicaciones mediante peticiones HTTP (GET, POST, etc.), y cómo pueden consumir o crear una API básica usando Node.js y Express.



¿Qué es una API?


API = Application Programming Interface

Es un conjunto de reglas que permite que dos sistemas se comuniquen entre sí.

Ejemplo cotidiano:

- Una aplicación de clima consulta una API del servicio meteorológico.
- Tu app de celular usa la API de Google Maps para mostrar direcciones.

¿Qué es una API REST?



Una API REST utiliza el protocolo HTTP (como los navegadores) para enviar y recibir información.

Operación	Método HTTP	Descripción
GET	Obtener información	Ej. listar usuarios
POST	Enviar información	Ej. crear un usuario
PUT	Actualizar información	Ej. modificar un usuario
DELETE	Eliminar información	Ej. borrar un usuario

Ejemplo de una URL REST: GET <https://jsonplaceholder.typicode.com/users>

Qué es JSON?



- JSON (JavaScript Object Notation) es el formato estándar para intercambiar datos entre servidores y aplicaciones.

```
{  
  "nombre": "Manuel",  
  "edad": 20,  
  "correo": "manu@example.com"  
}
```



Por qué es tan importante JSON?

Es el lenguaje común entre sistemas

- Cuando una aplicación (como tu servidor en Node.js) se comunica con otra (como una API externa), intercambian información en formato JSON.



Facilita la interoperabilidad

- JSON no depende de ningún lenguaje de programación.
- Lo entienden Python, Java, C#, PHP, Go, etc.
- Gracias a eso, puedes crear una API en Node.js y consumirla desde una app móvil, una página web o incluso un sistema embebido.

Es ligero y fácil de leer

- JSON usa una estructura muy simple basada en clave → valor, lo que lo hace ideal para enviar datos rápidamente por internet.
- Es mucho más ligero que otros formatos como XML.

Ejemplo Práctico



1. Crear una carpeta de trabajo

```
mkdir api-consumo
```

```
cd api-consumo
```

2. Inicializar el proyecto Node

```
npm init -y
```


Ejemplo Práctico



3. Instalar dependencias

Instalar Express y Axios

```
npm install express axios
```

Express: para crear el servidor web.

Axios: para consumir APIs externas fácilmente.



Crear el servidor

api-consumo/ [server.js](#) // Consumir API desde aquí



```
// Importar librerías
```

```
const express = require("express");
```

```
const axios = require("axios");
```

```
// Crear aplicación Express
```

```
const app = express();
```

```
const PORT = 3000;
```

```
// Ruta de inicio
```

```
app.get("/", (req, res) => {
```

```
  res.json({ mensaje: "Bienvenido al proyecto de consumo de APIs con Node.js y  
Express" });
```

```
});
```

```
// Ruta para consumir una API pública
```

```
app.get("/usuarios", async (req, res) => {
```

```
  try {
```

```
    // Consumir la API externa
```

```
    const respuesta = await axios.get("https://jsonplaceholder.typicode.com/users");
```



```
// Procesar datos: solo nombre y correo
const datos = respuesta.data.map(usuario => ({
  nombre: usuario.name,
  correo: usuario.email
}));

// Enviar respuesta al cliente
res.json(datos);
} catch (error) {
  console.error(error);
  res.status(500).json({ error: "Error al obtener los datos de la API" });
}
});

// Iniciar el servidor
app.listen(PORT, () => {
  console.log(`Servidor ejecutándose en http://localhost:${PORT}`);
});
```



Correr server

node [server.js](#)

Visualizar: localhost