

Node Js - JSON API

Topics : [Node js](#)

Written on [April 25, 2025](#)

What is a JSON API?

A **JSON API** is a type of **web API** (Application Programming Interface) that uses **JSON (JavaScript Object Notation)** as the format for **sending and receiving data** between a client (like a web app or mobile app) and a server.

How It Works:

- The **client** makes an HTTP request (like GET, POST, PUT, DELETE)
- The **server** responds with data in **JSON format**

Why JSON?

- Lightweight and fast
- Easy to read and write
- Works well with JavaScript

Commonly Used In:

- REST APIs
- Frontend-backend communication
- Mobile apps (iOS/Android)
- Microservices

Use Cases:

Action	Method	Endpoint	Description
Get all users	GET	/api/users	Fetch user list
Get single user	GET	/api/users/1	Fetch user with ID = 1
Add user	POST	/api/users	Add a new user
Update user	PUT	/api/users/1	Update user with ID = 1
Delete user	DELETE	/api/users/1	Delete user with ID = 1

Example :

```
const express = require('express');
const mysql = require('mysql2');

const app = express();

const connection = require('./db');

app.use(express.json());

// Fetch users
app.get('/api/users', (req, res) => {
  connection.query("SELECT * FROM users", (err, results) => {
    if (err) return res.status(500).send(err);
    res.json(results);
  });
});

app.get('/api/users/:id', (req, res) => {
  const user_id = req.params.id;
  connection.query("SELECT * FROM users where id = ?", [user_id], (err, results) => {
    if (err) return res.status(500).send(err);
    res.json(results);
  });
});

//DELETE user
app.delete('/api/users/:id', (req, res) => {
```

```
const user_id= req.params.id;

connection.query("delete FROM users where id= ?",[user_id], (err, results) => {

  if (err) return res.status(500).send(err);

  res.json({ message: 'User deleted' });

});

});

// Update user

app.put('/api/users/:id', (req, res) => {

  const user_id= req.params.id;

  connection.query("update users set name=? where id= ?",[req.body.name,user_id], (err, results)
=> {

    if (err) return res.status(500).send(err);

    res.json({ message: 'User updated' });

  });

});

// Insert user

app.post('/users', (req, res) => {

  const { name, email } = req.body;

  connection.query("INSERT INTO users (name, email) VALUES (?, ?)", [name, email], (err, result)
=> {

    if (err) return res.status(500).send(err);

    res.json({ message: "User added", id: result.insertId });

  });

});

app.listen(3000, () => console.log("Server running on port 3000"));
```

Request :

GET /api/users

Response:

[

```
{
  id: 3,
  name: "Smith Doe",
  email: "smith@example.com"
},
{
  id: 4,
  name: "smith doe",
  email: "smith@gmail.com"
},
{
  id: 5,
  name: "roberts joy",
  email: "robetrt@gmail.com"
},
{
  id: 6,
  name: "rose donaldo",
  email: "rose@gmail.com"
}
]
```