

Hướng Dẫn Xây Dựng API RESTful với Node.js và Express

Tác giả: Đặng Kim Thi

Mục Tiêu

Bài hướng dẫn này sẽ giúp bạn tạo một API RESTful hoàn chỉnh sử dụng **Node.js**, **Express**, và hệ thống quản lý file JSON thay vì cơ sở dữ liệu. Chúng ta sẽ:

- Cấu hình một server Express.
 - Xây dựng các endpoint để xử lý dữ liệu người dùng.
 - Sử dụng **fs (filesystem)** để lưu dữ liệu vào một file JSON.
 - Hiểu rõ các khái niệm về RESTful API, Express.js, Middleware, và cách quản lý dữ liệu JSON.
-

I. Các Công Nghệ Sử Dụng

1. Node.js

Là một môi trường chạy JavaScript phía server, cho phép chúng ta viết các ứng dụng backend bằng JS.

2. Express.js

Là một framework nhẹ của Node.js giúp xây dựng API dễ dàng.

3. fs (Filesystem module)

Là một module trong Node.js cho phép đọc, ghi file, giúp lưu dữ liệu vào JSON thay vì database.

4. RESTful API

Là một tiêu chuẩn thiết kế API giúp giao tiếp giữa client và server một cách hiệu quả.

II. Cài Đặt Môi Trường

Bước 1: Cài Node.js

Truy cập <https://nodejs.org/> để tải và cài đặt phiên bản mới nhất.

Bước 2: Tạo một thư mục dự án

```
mkdir my-api  
cd my-api
```

Bước 3: Khởi tạo Node.js project

```
npm init -y
```

Lệnh này sẽ tạo một file `package.json` chứa thông tin dự án.

Bước 4: Cài đặt Express

```
npm install express
```

Bước 5: Cài đặt fs module (có sẵn trong Node.js, không cần cài)

III. Xây Dựng API Step by Step

1. Khởi Tạo Server với Express

Giải thích:

- `express()` khởi tạo một ứng dụng Express.
- `app.use(express.json())` cho phép xử lý request body ở dạng JSON.
- `app.get('/', ...)` tạo route gốc trả về một thông điệp đơn giản.
- `app.listen(PORT, ...)` giúp ứng dụng lắng nghe trên một cổng cụ thể.

Code:

```
const express = require('express');
const app = express();
const PORT = 3000;

app.use(express.json());

app.get('/', (req, res) => {
  res.send('Chào mừng đến với API của chúng tôi!');
});

app.listen(PORT, () => {
  console.log(`Server chạy tại http://localhost:${PORT}`);
});
```

2. Quản Lý Dữ Liệu JSON

Giải thích:

- `fs.readFile` đọc dữ liệu từ file JSON.
- `fs.writeFile` ghi dữ liệu vào file JSON.

- `JSON.parse()` chuyển đổi dữ liệu JSON thành Object.
- `JSON.stringify()` chuyển đổi Object thành chuỗi JSON.

Code:

```
const fs = require('fs').promises;
const path = require('path');
const DATA_FILE = path.join(__dirname, 'data', 'data.json');

const readData = async () => {
  try {
    const data = await fs.readFile(DATA_FILE, 'utf8');
    return JSON.parse(data);
  } catch (err) {
    return [];
  }
};

const writeData = async (data) => {
  await fs.writeFile(DATA_FILE, JSON.stringify(data, null, 2), 'utf8');
};
```

3. Các API Endpoint

Lấy toàn bộ dữ liệu

```
app.get('/data', async (req, res) => {
  const data = await readData();
  res.json(data);
});
```

Giải thích:

- Lấy toàn bộ dữ liệu từ `data.json` và gửi về dưới dạng JSON.

Thêm dữ liệu mới

```
app.post('/data', async (req, res) => {
  const newData = req.body;
  const data = await readData();
  const newId = data.length > 0 ? Math.max(...data.map(item => item.id))
+ 1 : 1;
  const newItem = { id: newId, ...newData };
  data.push(newItem);
  await writeData(data);
});
```

```
res.json({ message: 'Ghi dữ liệu thành công', data: newItem });
});
```

Giải thích:

- Nhận dữ liệu mới từ request body.
- Tạo **id** mới cho dữ liệu.
- Lưu dữ liệu vào **data.json** và gửi phản hồi.

Cập nhật dữ liệu

```
app.put('/data/:id', async (req, res) => {
  const id = parseInt(req.params.id);
  const updatedData = req.body;
  const data = await readData();
  const index = data.findIndex(item => item.id === id);
  if (index === -1) {
    return res.status(404).json({ message: 'Không tìm thấy dữ liệu'
  });
  }
  data[index] = { ...data[index], ...updatedData };
  await writeData(data);
  res.json({ message: 'Cập nhật thành công', data: data[index] });
});
```

Giải thích:

- Tìm dữ liệu cần cập nhật bằng **id**.
- Nếu không tìm thấy, trả về lỗi **404**.
- Cập nhật dữ liệu và ghi vào file JSON.

Xóa dữ liệu

```
app.delete('/data/:id', async (req, res) => {
  const id = parseInt(req.params.id);
  const data = await readData();
  const newData = data.filter(item => item.id !== id);
  if (data.length === newData.length) {
    return res.status(404).json({ message: 'Không tìm thấy dữ liệu'
  });
  }
  await writeData(newData);
  res.json({ message: 'Xóa thành công' });
});
```

Giải thích:

- Lọc danh sách để loại bỏ phần tử có `id` tương ứng.
 - Nếu không tìm thấy, trả về lỗi `404`.
 - Ghi dữ liệu mới vào file JSON.
-

IV. Kết Luận

Bạn đã hoàn thành việc xây dựng API RESTful đơn giản sử dụng Node.js, Express và lưu dữ liệu vào file JSON.

Chúc bạn thành công!