Tổng Hợp Kiến Thức (1)

Thiết lập Môi trường

1. **Cài đặt NodeJS và npm/yarn:** Kiểm tra bằng `node -v` và `npm -v` (hoặc `yarn -v`).
2. **Tạo thư mục dự án:** `mkdir basic-express-api && cd basic-express-api`
3. **Khởi tạo dự án NodeJS:** `npm init -y` (tạo `package.json`).
4. **Cài đặt ExpressJS:** `npm install express`.
5. **Tạo file server chính (`server.js`):**

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

app.get('/', (req, res) => {
  res.send('Hello World from Express!');
});

app.listen(port, () => {
  console.log(`Server listening at <http://localhost>:${port}`);
});
```

6. **Chạy server:** `node server.js` hoặc `npm run dev` (sau khi cấu hình nodemon).
7. **Kiểm tra:** Truy cập `http://localhost:3000` trong trình duyệt.

Cải thiện Quy trình Phát triển & RESTful API

1. **Nodemon:** Tự động restart server khi code thay đổi.
   - Cài đặt: `npm install --save-dev nodemon`.
   - Thêm vào `scripts` trong `package.json`:

```
"scripts": {
  "start": "node server.js",
  "dev": "nodemon server.js"
}
```

- Chạy: `npm run dev`.

2. **RESTful API - 3 Quy tắc chính:**
   - **1. Tài nguyên (Resource) và URL (Địa chỉ):**
     - URL rõ ràng, dễ đoán, phản ánh tài nguyên.
     - Ví dụ:
       - `/products` (danh sách sản phẩm)
       - `/products/123` (sản phẩm ID 123)
   - **2. Hành động (Action) và HTTP Methods (Động từ):**
     - **GET:** Lấy dữ liệu (Read).
     - **POST:** Tạo dữ liệu (Create).
     - **PUT/PATCH:** Cập nhật dữ liệu (Update). PUT thay thế toàn bộ, PATCH cập nhật một phần.
     - **DELETE:** Xóa dữ liệu (Delete).
   - **3. Ngôn ngữ chung (JSON):**
     - Sử dụng JSON để trao đổi dữ liệu giữa client và server.

Xây dựng API - CRUD Operations

1. Read (GET)

Dữ liệu tạm (trong `server.js`):

```
let todos = [
  { id: 1, title: 'Learn Express', completed: false },
  { id: 2, title: 'Build API', completed: false },
```

```
  { id: 3, title: 'Test API', completed: false }
];
let nextId = 4;
```

a. Lấy tất cả Todos (`GET /todos`):

```
app.get('/todos', (req, res) => {
  res.status(200).json(todos);
});
```

b. Lấy một Todo theo ID (`GET /todos/:id`):

```
app.get('/todos/:id', (req, res) => {
  const todoId = parseInt(req.params.id);
  const todo = todos.find(t => t.id === todoId);

  if (todo) {
    res.status(200).json(todo);
  } else {
    res.status(404).json({ message: `Todo with id ${todoId} not found` });
  }
});
```

2. Create (POST)

Middleware (trong `server.js` , trước routes):

```
app.use(express.json()); // Parse JSON body
```

Endpoint: Tạo mới Todo (`POST /todos`):

```
app.post('/todos', (req, res) => {
  const { title } = req.body;

  if (!title) {
    return res.status(400).json({ message: 'Title is required' });
```

```
  }

  const newTodo = {
    id: nextId++,
    title: title,
    completed: false
  };

  todos.push(newTodo);

  res.status(201).json(newTodo);
});
```

3. Update (PUT)

Endpoint: Cập nhật Todo (`PUT /todos/:id`):

```
app.put('/todos/:id', (req, res) => {
  const todoId = parseInt(req.params.id);
  const { title, completed } = req.body;

  if (title === undefined || completed === undefined) {
     return res.status(400).json({ message: 'Both title and completed status are
required for PUT' });
  }
   if (typeof completed !== 'boolean') {
      return res.status(400).json({ message: 'Completed status must be a boole
an' });
  }

  const todoIndex = todos.findIndex(t => t.id === todoId);

  if (todoIndex === -1) {
    return res.status(404).json({ message: `Todo with id ${todoId} not found`
});
  }
```

```
  const updatedTodo = {
    id: todoId,
    title: title,
    completed: completed
  };

  todos[todoIndex] = updatedTodo;

  res.status(200).json(updatedTodo);
});
```

4. Delete (DELETE)

Endpoint: Xóa Todo (`DELETE /todos/:id`):

```
app.delete('/todos/:id', (req, res) => {
  const todoId = parseInt(req.params.id);
  const todoIndex = todos.findIndex(t => t.id === todoId);

  if (todoIndex === -1) {
    return res.status(404).json({ message: `Todo with id ${todoId} not found`
});
  }

  todos.splice(todoIndex, 1);

  res.status(204).send(); // 204 No Content
});
```