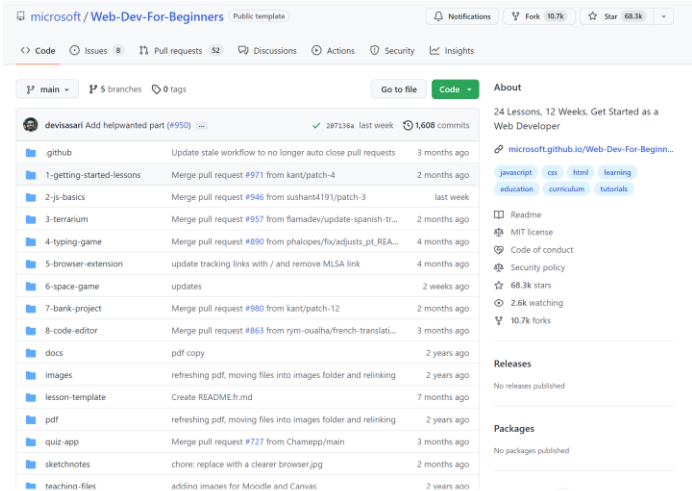


Hands on RestAPI with NodeJS

Mục tiêu:

1. Hiểu kiến trúc hướng dịch vụ: tương tác giữa client và server qua REST API
2. Vận dụng Git
3. Vận dụng NodeJS và Express
4. Sử dụng Rest Simple Client
5. Vận dụng Request trong REST API

Nguồn: <https://microsoft.github.io/Web-Dev-For-Beginners/#/>

Bước	Thao tác	Kiến thức/Đọc thêm
1.	<p>Khởi tạo dự án</p> <p>1.1. Tạo thư mục D:\learn_web</p> <p>1.2. Mở Visual Studio và mở thư mục vừa tạo</p> <p>1.3. Clone Repository mã nguồn: Mở terminal</p> <pre>> git clone https://github.com/microsoft/Web-Dev-For-Beginners</pre> <p>1.4. Truy cập trang github với trang trên, quan sát cấu trúc thư mục</p> 	<p>Q1. Git là gì?</p> <p>Q2. Giải thích câu lệnh git clone</p>
	1.5. Click vào project 7-bank-project	

	<div data-bbox="224 226 706 262" data-label="Section-Header"> <h3>3. Test Server với Simple Rest Client</h3> </div> <div data-bbox="297 300 795 333" data-label="Text"> <p>3.1 Cài đặt Simple Rest Client cho VS code</p> </div> <div data-bbox="297 371 1057 441" data-label="Text"> <p>3.2. Truy cập trang https://github.com/microsoft/Web-Dev-For-Beginners/tree/main/7-bank-project/api</p> </div> <div data-bbox="297 478 777 546" data-label="Text"> <p>3.3. Thử thực thi lệnh GET, với URL http://localhost:5000/api/acccount/test</p> </div> <div data-bbox="297 579 1112 1102" data-label="Image"> </div> <div data-bbox="297 1138 1097 1205" data-label="Text"> <p>3.3. Thử thực thi lệnh POST, tạo một account mới với thông tin bên dưới:</p> </div> <div data-bbox="297 1205 578 1701" data-label="Image"> </div> <div data-bbox="297 1732 941 1887" data-label="Text"> <p>URL: http://localhost:5000/api/accounts/ Header: { "Content-Type": "application/json"} Body: { "\"user\": \"baoan\", "\"currency\": \"\$\",</p> </div>	<div data-bbox="1141 226 1419 333" data-label="Text"> <p>Q7. Tìm hiểu cách thức đáp ứng của các câu lệnh GET phía server</p> </div> <div data-bbox="1141 352 1427 789" data-label="Code-Block"> <pre>router.get('/accounts/:user', (req, res) => { const account = db[req.params.user]; // Check if account exists if (!account) { return res.status(404).json({ error: 'User does not exist' }); } return res.json(account); });</pre> </div> <div data-bbox="1141 858 1419 966" data-label="Text"> <p>Q7. Tìm hiểu cách thức đáp ứng của các câu lệnh POST phía server</p> </div> <div data-bbox="1141 1035 1398 1104" data-label="Text"> <p>Q9. Tìm hiểu cấu trúc đối tượng JSON</p> </div>
--	---	--

	<pre>"description": "Test account", "balance": 75 }</pre> <p>3.4. Thử các câu lệnh khác: DELETE, PUT...</p>	
4.	<p>Thử ứng dụng với client app</p> <p>4.1. Mở terminal mới, chuyển vào thư mục solution của cùng project <code>> cd Web-Dev-For-Beginners/tree/main/7-bank-project/solution</code></p> <p>4.2. Chạy lite-server để chạy trang web phía client. <code>> npx lite-server</code></p> <p>4.3. Truy cập: http://localhost:3000, đăng nhập bằng username <i>test</i></p> <p>4.4. Thử chạy ứng dụng, tạo transaction mới, tạo user account mới</p> <p>4.3. Phân tích file index.html và file app.js (ChatGPT)</p> <p>4.5. So sánh việc chạy trên client (web) và test bằng Simple Rest Client</p>	<p>Q10. Tìm hiểu Express</p> <p>Q11. Tìm hiểu cách Server phân giải request từ Client và xử lý yêu cầu, gửi về cho client.</p>
5.	<p>Thử dockerize ứng dụng (tặng 01 điểm quá trình): Cho server và client chạy trên 02 container khác nhau, sử dụng cùng bộ cổng 5000 và 3000.</p>	
6.	<p>Thử thay thế phần cơ sở dữ liệu bằng MongoDB, dockerize lại ứng dụng phía server (tặng 2 điểm quá trình)</p>	