This documentation provides a step-by-step guide on how to create a CRUD (Create, Read, Update and Delete) application that focuses on managing a family tree using ExpressJS, NodeJS, Postman, and MongoDB as the database. The CRUD app allows users to add, view, update, and delete family members, including the father, mother, son, and daughter.

1. Setting up the Environment:

Before starting, ensure that you have Node.js and MongoDB installed on your system. Follow these steps to set up the environment:

a. Initialize the project:

- Create a new directory for your project and navigate into it.
- Open a terminal or command prompt and run the command: npm init -y

b. Install dependencies:

- Install ExpressJS, MongoDB driver, and other necessary packages by running: npm install express and npn install mongoose

c. Set up the MongoDB connection:

- Create a new file named index.js and establish a connection to the MongoDB database using the MongoDB driver.

2. Creating the API Endpoints: To implement the CRUD operations for managing family members, create the following API endpoints:

a. Create a Family Member:

- Use the ExpressJS router to handle a POST request to the endpoint ''localhost:4000/family/addFamily'.
- In the route handler, extract the family member data from the request body and insert it into the MongoDB collection.
- Respond with a success message or appropriate error response.

b. Retrieve Family Members:

- Handle a GET request to the endpoint 'localhost:4000/family/allFamily'.
- Fetch all family members from the MongoDB collection and send them as a response.

c. Update a Family Member:

- Handle a PUT request to the endpoint 'localhost:4000/family/updateFamily/:id'.
- Extract the family member ID from the request parameters and update the corresponding document in the MongoDB collection with the new data.
- Respond with a success message or appropriate error response.

d. Delete a Family Member:

- Handle a DELETE request to the endpoint 'localhost:4000/family/deleteFamily/:id'.
- Extract the family member ID from the request parameters and delete the corresponding document from the MongoDB collection.
- Respond with a success message or appropriate error response.

3. Testing the API using Postman: To verify the functionality of the CRUD app, use Postman to send requests to the API endpoints:

a. Open Postman and create a new request.

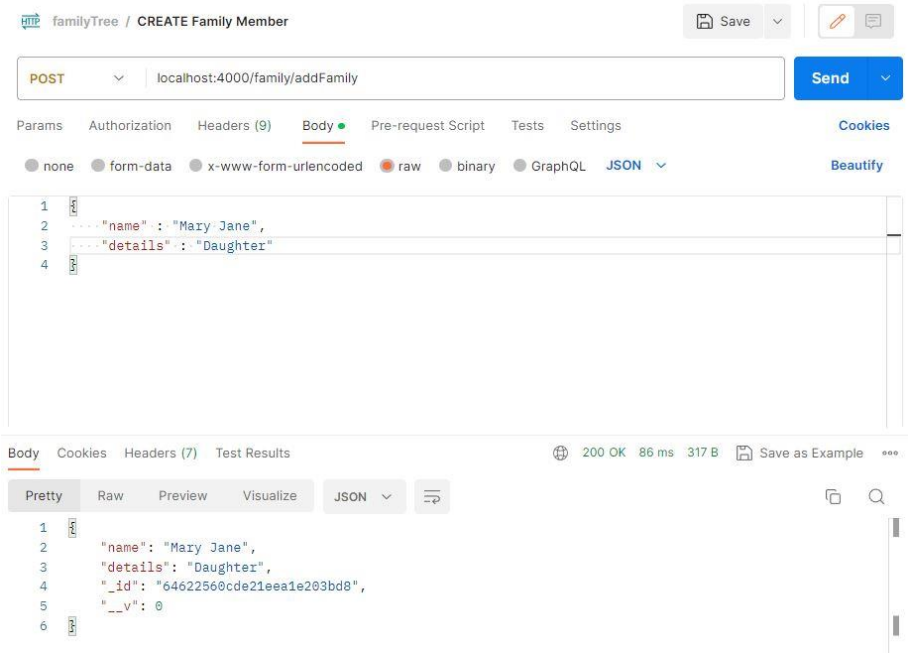
b. Set the request method to POST, GET, PUT, or DELETE, depending on the desired operation.

c. Enter the URL of the API endpoint (e.g., http://localhost:4000/family) and add any required request parameters or body data.

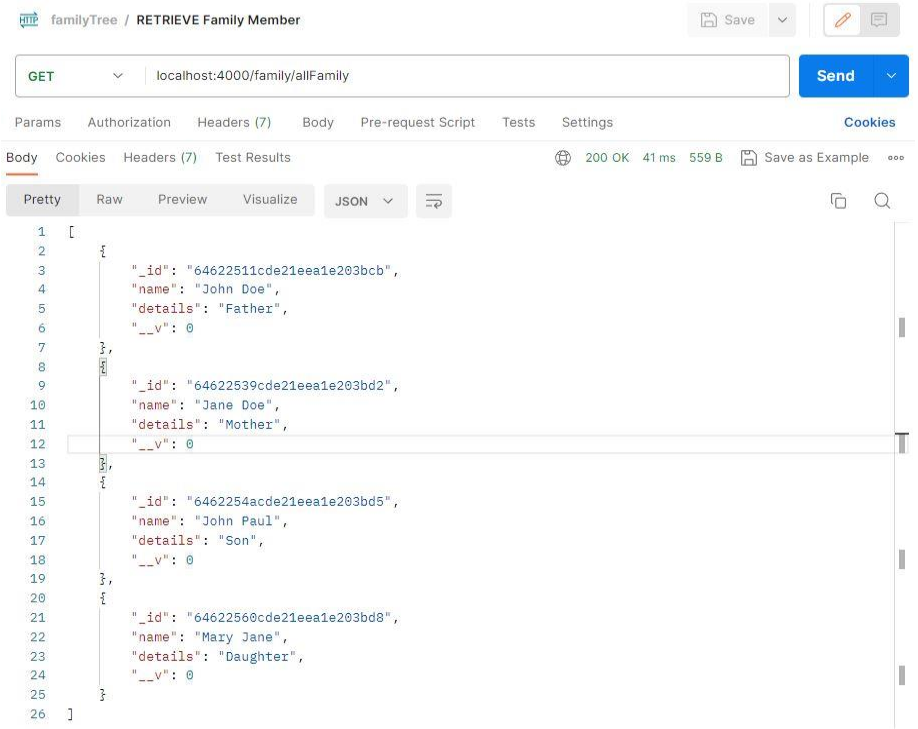
d. Click Send to execute the request and observe the response from the server.

POSTMAN

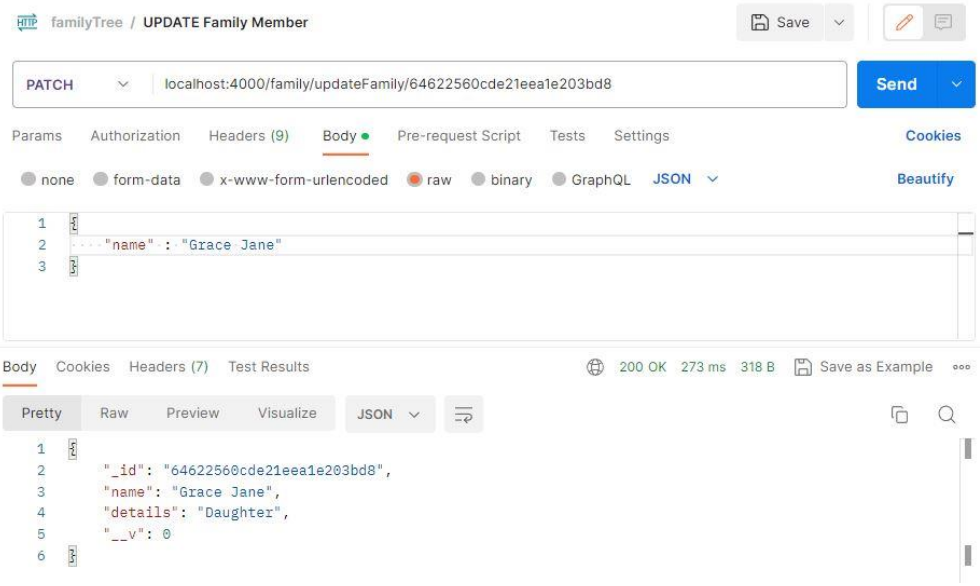
a. Create a Family Member



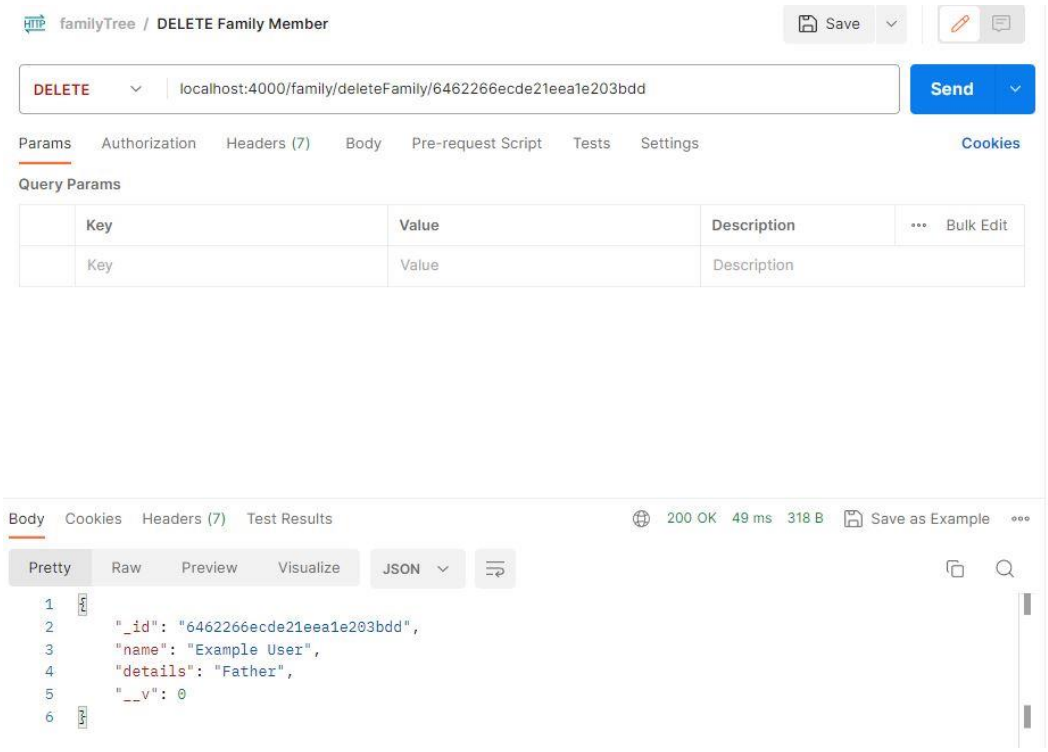
b. Retrieve Family Member



c. Update a Family Member



d. Delete a Family Member



MongoDB as database