

Comments:

I developed the entire project from scratch using resources like YouTube tutorials and documentation. Debugging was particularly interesting, as I learned to check API calls thoroughly, enhancing my problem-solving skills. I also faced challenges with data handling between the frontend and backend, which deepened my understanding of both Angular and Node.js.

Account Management Application

Full Stack + Angular 18 Material + Node JS + PostgreSQL

1. Database Setup

1.1. Creating the Database and Table

1. **Environment:** Use XAMPP to manage PostgreSQL.
2. **Database Creation:** Create a new PostgreSQL database to store account data.
3. **Table Structure:** Design a table to hold account details such as ID, name, email, and balance.

2. Backend Development

2.1. Setting Up Node.js

1. **Project Initialization:** Set up a Node.js project to handle server-side logic.
2. **API Implementation:** Develop RESTful APIs to manage account data. This includes endpoints for creating, reading, updating, and deleting account information.

2.2. API Testing

1. **Tool:** Use Postman to test the functionality of the implemented APIs.
2. **Database Verification:** Confirm that the changes reflect in the PostgreSQL database after API calls.

3. Frontend Development

3.1. Angular Application Setup

1. **Project Creation:** Set up an Angular standalone application using version 18.

2. **Material Design:** Install Angular Material for building a responsive and visually appealing user interface.

3.2. Proxy Configuration

1. **Setup:** Configure a proxy to allow the Angular application to communicate with the backend server running on a different port.

3.3. Component Generation

1. **Components:** Generate necessary components, services, and modules to structure the application effectively.
2. **Routing:** Implement routing logic to navigate between different components of the application.

3.4. Angular Material Table

1. **Data Display:** Create an Angular Material table to display account data, including actions for editing and deleting entries.
2. **HTTP Methods:** Implement the necessary HTTP methods (GET, POST, PUT, DELETE) in the service to manage account information.

3.5. Features Implementation

1. **Searching, Sorting, and Pagination:** Integrate functionalities to search, sort, and paginate through the account data displayed in the table.
2. **User Input Forms:** Use Angular Material form fields to capture user input for creating and editing accounts.

4. User Interaction

4.1. Form Handling

1. **Data Submission:** Capture input data from forms and submit it to the backend when creating or updating an account.
2. **Edit Functionality:** Enable users to populate form fields with existing account data for modification.

4.2. Account Deletion


1. **Removal Process:** Implement a feature to delete accounts from the database, ensuring that the table reflects the changes immediately.


References:

1. <https://www.youtube.com/watch?v=xjgCkEfUlwE>
2. <https://material.angular.io/>

Images:

Full Stack with Angular 18 Material Node JS & PostgreSQL





Account ID ↑	Name	Type	Balance	Edit	Delete
11	Jeff	Current	25000	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>


Items per page:

6 - 6 of 6

Insertion:

- Inserted Raju

Full Stack with Angular 18 Material Node JS & PostgreSQL


Enter name


Enter type

Enter balance

Submit

Cancel

Search account



Account ID ↑	Name	Type	Balance	Edit	Delete
11	Jeff	Current	25000	<div>Edit</div>	<div>Delete</div>
12	Raju	Current	5025	<div>Edit</div>	<div>Delete</div>

Items per page: 5 6 - 7 of 7 |< < > >|

Search:

- Search seshu

Full Stack with Angular 18 Material Node JS & PostgreSQL


Enter name


Enter type

Enter balance

Submit

Cancel

Sesh



Account ID ↑	Name	Type	Balance	Edit	Delete
1	Seshu Kolli	Savings	1000	<div>Edit</div>	<div>Delete</div>

Items per page: 5 6 - 7 of 7 |< < > >|


Edit:

- Edited seshu kolli to Babu

Full Stack with Angular 18 Material Node JS & PostgreSQL

Enter name 	Enter type	Enter balance
---	------------	---------------

Submit	Cancel
--------	--------

Search account


Account ID	Name	Type	Balance	Edit	Delete
12	Raju	Current	5025	Edit	Delete
1	Babu	Current	11000	Edit	Delete

Items per page:

5


6 – 7 of 7

|< < > >|

Delete:

Deleted Raju


Full Stack with Angular 18 Material Node JS & PostgreSQL







Submit

Cancel

Raju



Account ID	Name	Type	Balance	Edit	Delete
Items per page:			5	6 – 6 of 6	   

Postgres DataBase:

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- Publications
- Schemas (2)
 - acc
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (1)
 - account
 - Columns (4)
 - id
 - name
 - type
 - balance
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers

Dependencies X Dependents X Processes X account_master/postgres@PostgreSQL 17*

account_master/postgres@PostgreSQL 17

Query Query History Scratch Pad X

```
1 SELECT id, name, type, balance
2 FROM acc.account;
```

Data Output Messages Notifications

	id	name	type	balance
	[PK] integer	character varying (250)	character varying (250)	bigint
1	9	Sarah Davis	Investment	5000
2	10	Sam	Expenses	25000
3	2	kevin1	Savings	5500
4	7	Emily Johnson1	Business	7500
5	11	Jeff	Current	25000
6	1	Babu	Current	11000

Total rows: 6 of 6 Query complete 00:00:00.224 Ln 2, Col 19

Postman:

HomeWorkspacesAPI Network

Search Postman

Invite

Upgrade

My Workspace

NewImport

Overview

POST http://localhost:3005/api/v1/accounts

GET http://localhost:3005/api/v1/accounts

PUT http://localhost:3005/api/v1/accounts

PUT http://localhost:3005/api/v1/accounts

DEL http://localhost:3005/api/v1/accounts

+

No environment

REST API basics: CRUD, test & variable

Environments

History

http://localhost:3005/api/v1/accounts

Save

Share

GET

http://localhost:3005/api/v1/accounts

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body

Cookies

Headers (0)

Test Results

200 OK

80 ms

640 B

Pretty

Raw

Preview

Visualize

JSON

```
1 [
2   {
3     "id": 9,
4     "name": "Sarah Davis",
5     "type": "Investment",
6     "balance": "50000"
7   },
8   {
9     "id": 10,
10    "name": "Sam",
11    "type": "Expenses",
12    "balance": "25000"
13  },
14  {
15    "id": 2,
16    "name": "Kevin1",
17    "type": "Savings",
18    "balance": "55000"
19  },
20 ]
```

Online

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Vault

Trash