

Backend Developer Assignment

Introduction: Building a RESTful API for a task management system using Node.js, TypeScript, PostgreSQL, Factly and Firebase Functions.

API Design and Routes: Design and implement API routes for the following functionalities:

1. Create a new task

URL: <http://localhost:3000/tasks>

Request Type: POST

BODY:

```
{
  "title": "walk",
  "description": "at 6:00 am daily"
}
```

REQUEST:

The screenshot displays a REST client interface with a top navigation bar containing tabs for different HTTP methods: DEL, POST, PUT, GET, and DELETE. The main workspace is titled 'REST API basics: CRUD, test & variable / Post data'. The 'POST' method is selected, and the URL is set to 'http://localhost:3000/tasks'. The 'Body' tab is active, showing a JSON payload:

```
{
  "title": "Morning Walk",
  "description": "Encourage physical activity and promote a healthy lifestyle by going for a morning walk."
}
```

. The 'Send' button is visible. Below the workspace, the 'Test Results' tab shows a successful response with status '200 OK', time '167 ms', and size '308 B'. The response body is displayed in a 'Pretty' JSON format:

```
{
  "id": 6,
  "title": "Morning Walk",
  "description": "Encourage physical activity and promote a healthy lifestyle by going for a morning walk."
}
```

. The bottom of the image shows a Windows taskbar with various application icons and a system clock indicating 3:54 PM on 3/14/2024.

RESPONSE:

The screenshot shows a PostgreSQL database interface. On the left, the Object Explorer displays the database structure, including a table named 'tasks'. The main query editor shows the following SQL script:

```
1 -- Database: apidata
2
3 -- DROP DATABASE IF EXISTS apidata;
4
5 CREATE DATABASE apidata
6 WITH
7     OWNER = postgres
8     ENCODING = 'UTF8'
9     LC_COLLATE = 'English_United States.1252'
10    LC_CTYPE = 'English_United States.1252'
11    LOCALE_PROVIDER = 'libc'
12    TABLESPACE = pg_default
13    CONNECTION LIMIT = -1
14    IS_TEMPLATE = False;
15
16
17 select * from tasks;
```

The 'Data Output' tab shows the results of the query:

id	title	description
1	6	Morning Walk

The description for the task is: "Encourage physical activity and promote a healthy lifestyle by going for a morning walk."

2. Retrieve a task by ID

URL: <http://localhost:3000/tasks/6>

Request Type: GET

REQUEST/RESPONSE:

The screenshot shows a REST client interface with a GET request to `http://localhost:3000/tasks/6`. The response is a JSON object:

```
1 {
2   "id": 6,
3   "title": "Morning Walk",
4   "description": "Encourage physical activity and promote a healthy lifestyle by going for a morning walk."
5 }
```

The status of the request is 200 OK, with a response time of 36 ms and a size of 308 B.

3. List all tasks

URL: <http://localhost:3000/tasks>

Request Type: GET

REQUEST/RESPONSE:

The screenshot shows a REST client interface with the following details:

- URL:** <http://localhost:3000/tasks>
- Request Type:** GET
- Method:** GET
- Body:** This request does not have a body.
- Status:** 200 OK
- Time:** 33 ms
- Size:** 764 B
- Response Format:** JSON
- Response Data:**

```
[{"id": 6, "title": "Morning Walk", "description": "Encourage physical activity and promote a healthy lifestyle by going for a morning walk."}, {"id": 7, "title": "Daily Reading Habit", "description": "Cultivate a habit of reading regularly to expand knowledge, stimulate creativity, and reduce stress."}, {"id": 8, "title": "Home Organization", "description": "Streamline and tidy up living spaces to create a more functional and aesthetically pleasing environment."}, {"id": 9, "title": "Self-Care Routine", "description": "Prioritize self-care activities to nurture physical, mental, and emotional well-being."}]
```

4. Update a task

URL: <http://localhost:3000/tasks/9>

Request Type: PUT

BEFORE UPDATE:

The screenshot shows the PostgreSQL Enterprise Studio interface. The left pane displays the 'Object Explorer' with a tree view of the database structure, including 'Servers (1)', 'PostgreSQL 16', 'Databases (2)', and 'apidata'. The main pane shows the 'Query' editor with the following SQL script:

```
-- Database: apidata
-- DROP DATABASE IF EXISTS apidata;
CREATE DATABASE apidata
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'English_United States.1252'
  LC_CTYPE = 'English_United States.1252'
  LOCALE_PROVIDER = 'libc'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;

select * from tasks;
```

The 'Data Output' pane shows the results of the query, displaying a table with 4 rows and 3 columns: 'id' (integer), 'title' (character varying), and 'description' (text).

id	title	description
1	6 Morning Walk	Encourage physical activity and promote a healthy lifestyle by going for a morning walk.
2	7 Daily Reading Habit	Cultivate a habit of reading regularly to expand knowledge, stimulate creativity, and reduce stress.
3	8 Home Organization	Streamline and tidy up living spaces to create a more functional and aesthetically pleasing environ...
4	9 Self-Care Routine	Prioritize self-care activities to nurture physical, mental, and emotional well-being.

The status bar at the bottom indicates 'Total rows: 4 of 4' and 'Query complete 00:00:00.077'.

PUT REQUEST:

The screenshot shows the Postman interface for a PUT request. The URL is 'http://localhost:3000/tasks/9'. The request body is a JSON object:

```
{
  "id": 9,
  "title": "Fitness Routine",
  "description": "Establish a Weekly Exercise Plan"
}
```

The response status is '200 OK' with a time of '34 ms' and a size of '254 B'. The response body is a JSON object:

```
{
  "id": 9,
  "title": "Fitness Routine",
  "description": "Establish a Weekly Exercise Plan"
}
```

AFTER UPDATE:

The screenshot shows a PostgreSQL client window with the title 'apidata/postgres@PostgreSQL 16*'. The 'Query' tab is active, displaying the following SQL script:

```
1 -- Database: apidata
2
3 -- DROP DATABASE IF EXISTS apidata;
4
5 CREATE DATABASE apidata
6 WITH
7 OWNER = postgres
8 ENCODING = 'UTF8'
9 LC_COLLATE = 'English_United_States.1252'
10 LC_CTYPE = 'English_United_States.1252'
11 LOCALE_PROVIDER = 'libc'
12 TABLESPACE = pg_default
13 CONNECTION LIMIT = -1
14 IS_TEMPLATE = False;
15
16
17 select * from tasks;
```

The 'Data Output' tab is also active, showing the results of the query in a table with 4 rows and 3 columns: id, title, and description.

id	title	description
6	Morning Walk	Encourage physical activity and promote a healthy lifestyle by going for a morning walk.
7	Daily Reading Habit	Cultivate a habit of reading regularly to expand knowledge, stimulate creativity, and reduce stress.
8	Home Organization	Streamline and tidy up living spaces to create a more functional and aesthetically pleasing environm...
9	Fitness Routine	Establish a Weekly Exercise Plan

The status bar at the bottom indicates 'Total rows: 4 of 4', 'Query complete 00:00:00.100', and 'Ln 17, Col 1'. The system clock shows 4:14 PM on 3/14/2024.

5. Delete a task

URL: <http://localhost:3000/tasks/9>

Request Type: PUT

The screenshot shows a REST client interface with the URL 'http://localhost:3000/tasks/9' and the request type set to 'DELETE'. The 'Send' button is visible. The 'Body' tab is active, showing the response in JSON format:

```
{
  "id": 9,
  "title": "Fitness Routine",
  "description": "Establish a Weekly Exercise Plan"
}
```

The status bar at the bottom indicates 'Status: 200 OK', 'Time: 35 ms', and 'Size: 254 B'. The system clock shows 4:16 PM on 3/14/2024.

DATABASE:

The screenshot shows a PostgreSQL database management tool interface. On the left, the 'Object Explorer' pane displays the database structure, including 'Servers (1)', 'PostgreSQL 16', 'Databases (2)', and 'apidata'. The 'apidata' database is expanded, showing 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas', 'Subscriptions', 'postgres', 'Login/Group Roles', and 'Tablespaces (2)'. The 'Tablespaces (2)' are 'pg_default' and 'pg_global'.

The main pane shows the 'Query' editor with the following SQL code:

```
-- Database: apidata
-- DROP DATABASE IF EXISTS apidata;

CREATE DATABASE apidata
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'English_United_States.1252'
  LC_CTYPE = 'English_United_States.1252'
  LOCALE_PROVIDER = 'libc'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;

select * from tasks;
```

The 'Data Output' pane shows the results of the query, displaying a table with 3 rows and 3 columns: 'id', 'title', and 'description'.

id	title	description
1	6 Morning Walk	Encourage physical activity and promote a healthy lifestyle by going for a morning walk.
2	7 Daily Reading Habit	Cultivate a habit of reading regularly to expand knowledge, stimulate creativity, and reduce stress.
3	8 Home Organization	Streamline and tidy up living spaces to create a more functional and aesthetically pleasing environm...

The status bar at the bottom indicates 'Total rows: 3 of 3' and 'Query complete 00:00:00.060'. The bottom right corner shows the time '4:23 PM' and the date '3/14/2024'.

DATABASE: PostgreSQL

BACKEND: Node.js, TypeScript

API TESTING: POSTMAN

Validation and Error Handling:

- **npm install fastify-schema-** We import FastifySchema from fastify-schema to define JSON schemas for request validation