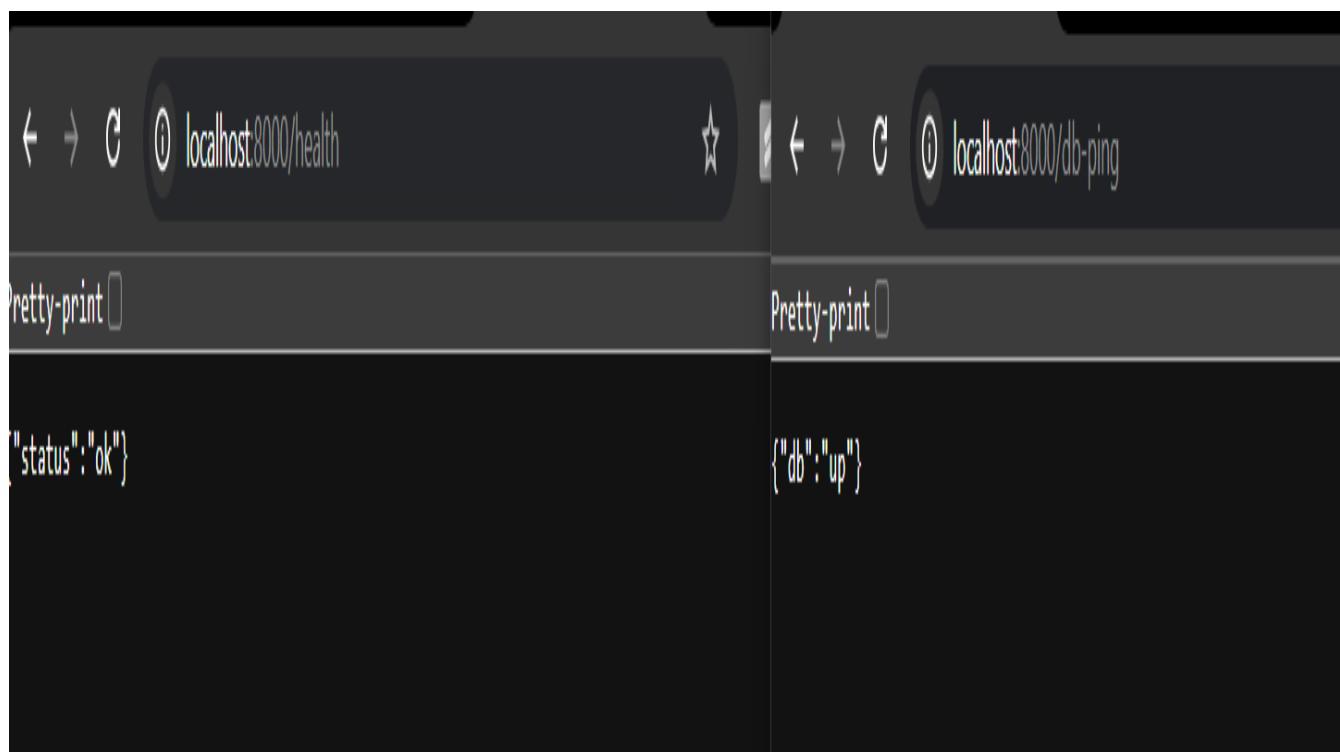


FastAPI + PostgreSQL Docker Compose Assignment

This document shows successful FastAPI and PostgreSQL integration using Docker Compose, with SQL operations executed in pgAdmin.

FastAPI Health & DB Ping



The screenshot shows a pgAdmin interface with two database connections:

- localhost:8000/health**: This connection shows a successful health check with the response: {"status": "ok"}.
- localhost:8000/db-ping**: This connection shows a successful database ping with the response: {"db": "up"}.

Create Table Query

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with the 'fastapi_db' database selected. The main pane shows a SQL query window with the following code:

```
6  );
7
8 v CREATE TABLE calculations (
9   id SERIAL PRIMARY KEY,
10  operation VARCHAR(20) NOT NULL,
11  operand_a FLOAT NOT NULL,
12  operand_b FLOAT NOT NULL,
13  result FLOAT NOT NULL,
14  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
15  user_id INTEGER NOT NULL,
16  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
17 );
18
```

The status bar at the bottom indicates "Query returned successfully in 103 msec."

Insert Query Execution

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the Object Explorer with the 'fastapi_db' database selected. The main pane shows a SQL query window with the following code:

```
1 v INSERT INTO users (username, email)
2 VALUES
3 ('alice', 'alice@example.com'),
4 ('bob', 'bob@example.com');
5
6 v INSERT INTO calculations (operation, operand_a, operand_b, result, user_id)
7 VALUES
8 ('add', 2, 3, 5, 1),
9 ('divide', 10, 2, 5, 1),
10 ('multiply', 4, 5, 20, 2);
11
```

The status bar at the bottom indicates "Query returned successfully in 46 msec."

Select Query Results

The screenshot shows the pgAdmin 4 interface connected to a PostgreSQL database named 'fastapi_db'. The left sidebar displays the Object Explorer with the 'fastapi_db' schema selected. The main area shows a SQL query window containing three SELECT statements. The third statement performs a JOIN between 'users' and 'calculations' tables. Below the query window is a Data Output tab displaying the results of the JOINed query.

```
-- All users
SELECT * FROM users;
-- All calculations
SELECT * FROM calculations;
-- Join users and calculations
SELECT u.username, c.operation, c.operand_a, c.operand_b, c.result
FROM calculations c
JOIN users u ON c.user_id = u.id;
```

	username	operation	operand_a	operand_b	result
1	alice	add		2	3
2	alice	divide		10	2
3	bob	multiply		4	5

Update Query Execution

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows a tree structure with Servers (1) > Local-DB > Databases (2) > fastapi_db.
- Query Editor:** Contains the following SQL code:

```
1 UPDATE calculations
2 SET result = 6
3 WHERE id = 1;
```
- Data Output:** Displays the message "Query returned successfully in 54 msec."

Update Query Results

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** Shows a tree structure with Servers (1) > Local-DB > Databases (2) > fastapi_db.
- Query Editor:** Contains the following SQL code:

```
1 SELECT * FROM calculations WHERE id = 1;
```
- Data Output:** Displays the results of the query in a table format:

	id [PK] integer	operation character varying (20)	operand_a double precision	operand_b double precision	result double precision	timestamp timestamp without time zone	user_id integer
1	1	add	2	3	6	2025-11-09 03:29:33.393987	1

Delete Query Execution

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'fastapi_db' database, the 'fastapi_db/postgres@Local-DB' connection is selected. The 'Query' tab in the main pane displays the following SQL code:

```
1 v DELETE FROM calculations
2 WHERE id = 2;
```

The 'Messages' tab at the bottom shows the result of the query:

DELETE 1
Query returned successfully in 53 msec.

Delete Query Results

The screenshot shows the pgAdmin interface for a PostgreSQL database named 'fastapi_db'. In the Object Explorer, under 'Databases (2)', 'fastapi_db' is selected. The 'Query' tab in the main pane contains the following SQL code:

```
1 SELECT * FROM calculations ORDER BY id;
```

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

	id [PK] integer	operation character varying (20)	operand_a double precision	operand_b double precision	result double precision	timestamp timestamp without time zone	user_id integer
1	1	add	2	3	6	2025-11-09 03:29:33.393987	1
2	3	multiply	4	5	20	2025-11-09 03:29:33.393987	2

Delete 2 (Cascade Test)

The screenshot shows the pgAdmin interface for the same PostgreSQL database 'fastapi_db'. In the Object Explorer, under 'Databases (2)', 'fastapi_db' is selected. The 'Query' tab in the main pane contains the following SQL code:

```
1 DELETE FROM users WHERE id = 1;
2 SELECT * FROM calculations ORDER BY id;
3
```

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

	id [PK] integer	operation character varying (20)	operand_a double precision	operand_b double precision	result double precision	timestamp timestamp without time zone	user_id integer
1	3	multiply	4	5	20	2025-11-09 03:29:33.393987	2

