

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

З лабораторної роботи № 2 з дисципліни  
«Технології розробки вбудованих IoT системи»

„CRUD ОПЕРАЦІЇ. РОЗРОБКА CRUD ОПЕРАЦІЙ.”

**Перевірив**

---

(прізвище, ім'я, по батькові)

Київ 2023

1) Посилання на git repo: <https://github.com/faaant/iot>

2) Реалізація CRUD операцій для `processed_agent_data`

```
# FastAPI CRUDL endpoints
@app.post("/processed_agent_data/")
async def create_processed_agent_data(data: List[ProcessedAgentData]):
    query_values = []
    for item in data:
        value = {
            "road_state": item.road_state,
            "x": item.agent_data.accelerometer.x,
            "y": item.agent_data.accelerometer.y,
            "z": item.agent_data.accelerometer.z,
            "latitude": item.agent_data.gps.latitude,
            "longitude": item.agent_data.gps.longitude,
            "timestamp": item.agent_data.timestamp,
            "user_id": item.agent_data.user_id,
        }
        query_values.append(value)

    query = processed_agent_data.insert().values(query_values)

    # Execute the query within a transaction
    with engine.begin() as conn:
        conn.execute(query)

    # Send data to subscribers
    for item in data:
        await send_data_to_subscribers(item.agent_data.user_id, item.model_dump())

@app.get(
    "/processed_agent_data/{processed_agent_data_id}",
    response_model=ProcessedAgentDataInDB,
)
def read_processed_agent_data(processed_agent_data_id: int):
    # Construct a query to select data from the processed_agent_data table based on
    the provided ID
    query = select(processed_agent_data).where(
        processed_agent_data.c.id == processed_agent_data_id
    )

    # Execute the query and retrieve the data
    with engine.connect() as conn:
        result = conn.execute(query)
        data = result.fetchone()
        if data is None:
            raise HTTPException(status_code=404, detail="Data not found")
    return data
```

```

@app.get("/processed_agent_data/", response_model=list[ProcessedAgentDataInDB])
def list_processed_agent_data():
    # Construct a query to select data from the processed_agent_data table
    query = select(processed_agent_data)

    # Execute the query and retrieve the data
    with engine.connect() as conn:
        result = conn.execute(query)
        data_list = result.fetchall()
    return data_list

@app.put(
    "/processed_agent_data/{processed_agent_data_id}",
    response_model=ProcessedAgentDataInDB,
)
def update_processed_agent_data(processed_agent_data_id: int, data:
ProcessedAgentData):
    # Construct the update query
    query = processed_agent_data.update().where(
        processed_agent_data.c.id == processed_agent_data_id
    ).values(
        road_state=data.road_state,
        x=data.agent_data.accelerometer.x,
        y=data.agent_data.accelerometer.y,
        z=data.agent_data.accelerometer.z,
        latitude=data.agent_data.gps.latitude,
        longitude=data.agent_data.gps.longitude,
        timestamp=data.agent_data.timestamp,
        user_id=data.agent_data.user_id,
    )

    # Execute the query within a transaction
    with engine.begin() as conn:
        result = conn.execute(query)

        # Check if any rows were affected (updated)
        if result.rowcount == 0:
            raise HTTPException(status_code=404, detail="Data not found")

    return read_processed_agent_data(processed_agent_data_id)

@app.delete(
    "/processed_agent_data/{processed_agent_data_id}",
    response_model=ProcessedAgentDataInDB,
)
def delete_processed_agent_data(processed_agent_data_id: int):
    processed_agent_data_to_be_removed =
read_processed_agent_data(processed_agent_data_id);

```

```

# Construct the delete query
query = processed_agent_data.delete().where(
    processed_agent_data.c.id == processed_agent_data_id
)

# Execute the query within a transaction
with engine.begin() as conn:
    result = conn.execute(query)

    # Check if any rows were affected (deleted)
    if result.rowcount == 0:
        raise HTTPException(status_code=404, detail="Data not found")

return processed_agent_data_to_be_removed

```

### 3) Показ результатів роботи в Swagger та PgAdmin

#### POST:

#### Swagger:

POST

/processed\_agent\_data/ Create Processed Agent Data

Parameters

No parameters

Request body required

application/json

```

[
  {
    "road_state": "state _-_:D",
    "agent_data": {
      "user_id": 123456789,
      "accelerometer": {
        "x": 1,
        "y": 2,
        "z": 3
      },
      "gps": {
        "latitude": 0,
        "longitude": 0
      },
      "timestamp": "2024-03-10T16:28:04.112Z"
    }
  }
]

```

Servers

These operation-level options override the global server options.

/

Execute

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/processed_agent_data/' \
  -H 'accept: application/json' \
  -H 'content-type: application/json' \
  -d '{
    "road_state": "state _- :D",
    "agent_data": {
      "user_id": 123456789,
      "accelerometer": {
        "x": 1,
        "y": 2,
        "z": 3
      },
      "gps": {
        "latitude": 0,
        "longitude": 0
      },
      "timestamp": "2024-03-10T16:28:04.112Z"
    }
  }'
```

Request URL

http://127.0.0.1:8000/processed\_agent\_data/

Server response

Code Details

200

Response body

null

Response headers

content-length: 4  
content-type: application/json  
date: Sun, 10 Mar 2024 16:29:10 GMT  
server: uvicorn

Результат у pgAdmin:

pgAdmin

Object Explorer

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (1)
      - processed\_agent\_data
    - Trigger Functions
    - Types
    - Views
  - Subscriptions
  - Login/Group Roles
  - Tablespaces

Query

```
1 SELECT * FROM public.processed_agent_data
2 ORDER BY id ASC
```

Data Output

id	road_state	user_id	x	y	z	latitude	longitude	timestamp
[PK] integer	character varying (255)	integer	double precision	double precision	double precision	double precision	double precision	timestamp without time zone
1	2	string	111	0	0	0	0	2024-03-10 15:08:53.776
2	3	state _- :D	123456789	1	2	3	0	2024-03-10 16:28:04.112

Total rows: 2 of 2 Query complete 00:00:00.339 Ln 1, Col 1

GET:

Swagger:

GET /processed\_agent\_data/ List Processed Agent Data

Parameters

No parameters

Servers

These operation-level options override the global server options.

/

Execute

Responses

Code	Description	Links
200	<div>Successful Response</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header</div> <div>Example Value   Schema</div> <pre>[   {     "id": 0,     "road_state": "string",     "user_id": 0,     "x": 0,     "y": 0,     "z": 0,     "latitude": 0,     "longitude": 0,     "timestamp": "2024-03-10T16:31:18.093Z"   } ]</pre>	No links

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/processed_agent_data/' \
  -H 'accept: application/json'
```

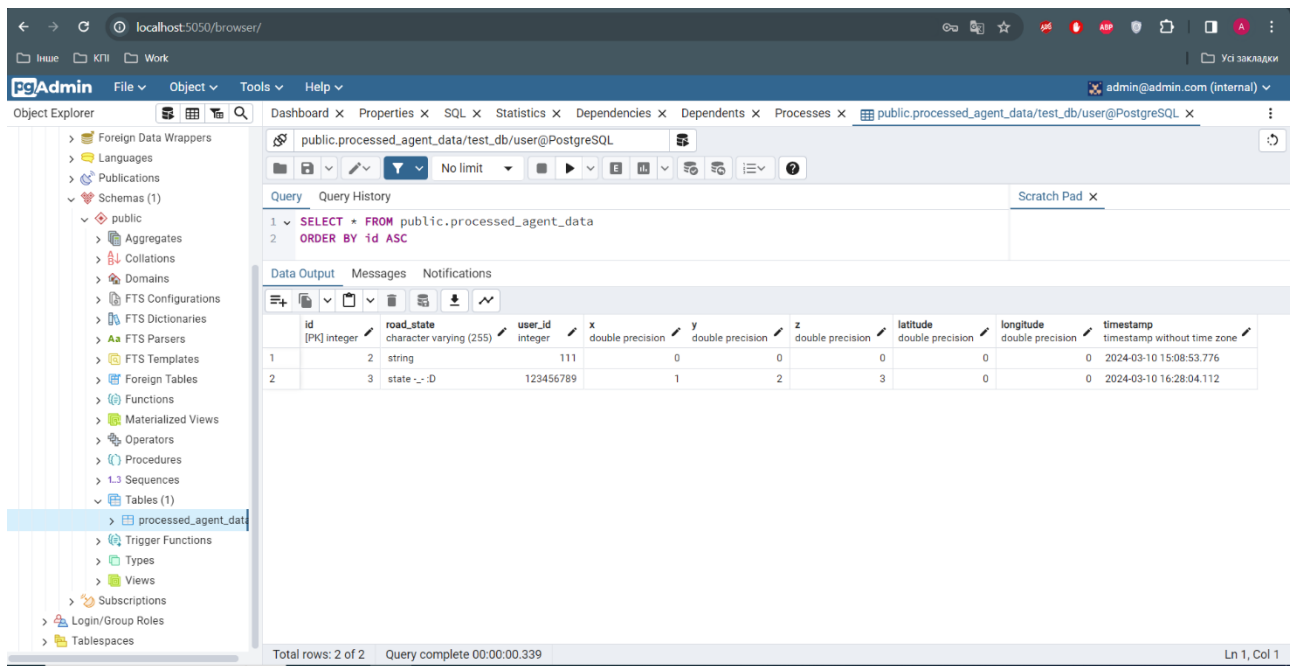
Request URL

```
http://127.0.0.1:8000/processed_agent_data/
```

Server response

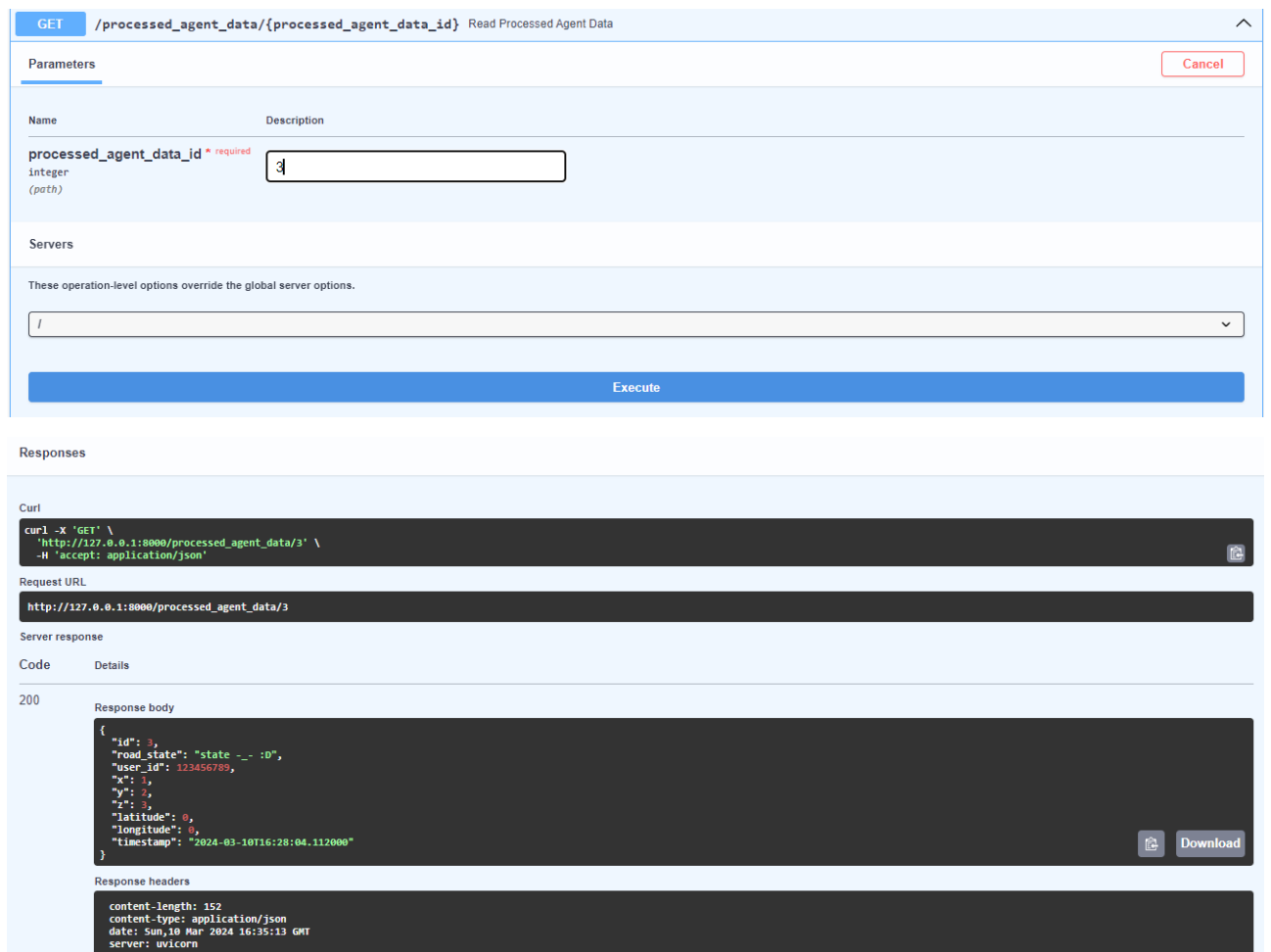
Code	Details
200	<div>Response body</div> <pre>[   {     "id": 2,     "road_state": "string",     "user_id": 111,     "x": 0,     "y": 0,     "z": 0,     "latitude": 0,     "longitude": 0,     "timestamp": "2024-03-10T15:08:53.776000"   },   {     "id": 3,     "road_state": "state _- :0",     "user_id": 123456789,     "x": 1,     "y": 2,     "z": 3,     "latitude": 0,     "longitude": 0,     "timestamp": "2024-03-10T16:28:04.112000"   } ]</pre> <div>Response headers</div> <pre>content-length: 295 content-type: application/json date: Sun, 10 Mar 2024 16:32:03 GMT server: uvicorn</pre>

PgAdmin:



GET:ID

Swagger:



## UPDATE:

## Swagger:

PUT

/processed\_agent\_data/{processed\_agent\_data\_id}

Update Processed Agent Data

Cancel

Reset

Parameters

Name	Description
<b>processed_agent_data_id</b> * required	
integer	
(path)	

3

Request body required

application/json

```
{  "road_state": "state _- :D",  "agent_data": {    "user_id": 123456789,    "accelerometer": {      "x": 10,      "y": 20,      "z": 30    },    "gps": {      "latitude": 50,      "longitude": 70    },    "timestamp": "2024-03-10T16:38:14.314Z"  }}
```

Servers

These operation-level options override the global server options.

/

Execute

Clear

### Responses

#### Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/processed_agent_data/3' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "road_state": "state _- :D",
    "agent_data": {
      "user_id": 123456789,
      "accelerometer": {
        "x": 10,
        "y": 20,
        "z": 30
      },
      "gps": {
        "latitude": 50,
        "longitude": 70
      },
      "timestamp": "2024-03-10T16:38:14.314Z"
    }
  }'
```

#### Request URL

http://127.0.0.1:8000/processed\_agent\_data/3

#### Server response

Code	Details
------	---------

200

#### Response body

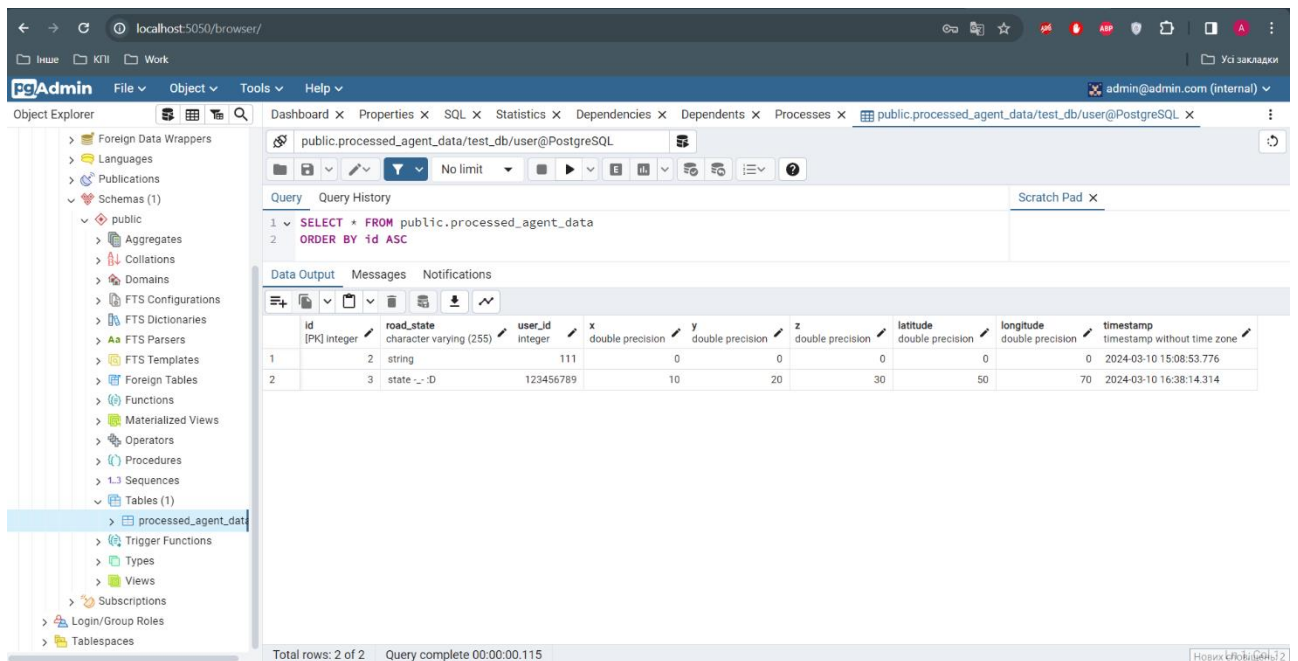
```
{  "id": 3,  "road_state": "state _- :D",  "user_id": 123456789,  "x": 10,  "y": 20,  "z": 30,  "latitude": 50,  "longitude": 70,  "timestamp": "2024-03-10T16:38:14.314000"}
```

#### Response headers

```
content-length: 157
content-type: application/json
date: Sun, 10 Mar 2024 16:39:27 GMT
server: uvicorn
```

## pgAdmin:





**DELETE:**

Swagger:

DELETE

/processed\_agent\_data/{processed\_agent\_data\_id} Delete Processed Agent Data

Parameters

Cancel

Name	Description
processed_agent_data_id * required integer (path)	<input type="text" value="2"/>

Servers

These operation-level options override the global server options.

/

Execute

## Responses

### Curl

```
curl -X 'DELETE' \
  'http://127.0.0.1:8000/processed_agent_data/2' \
  -H 'accept: application/json'
```

### Request URL

```
http://127.0.0.1:8000/processed_agent_data/2
```

### Server response

Code Details

200

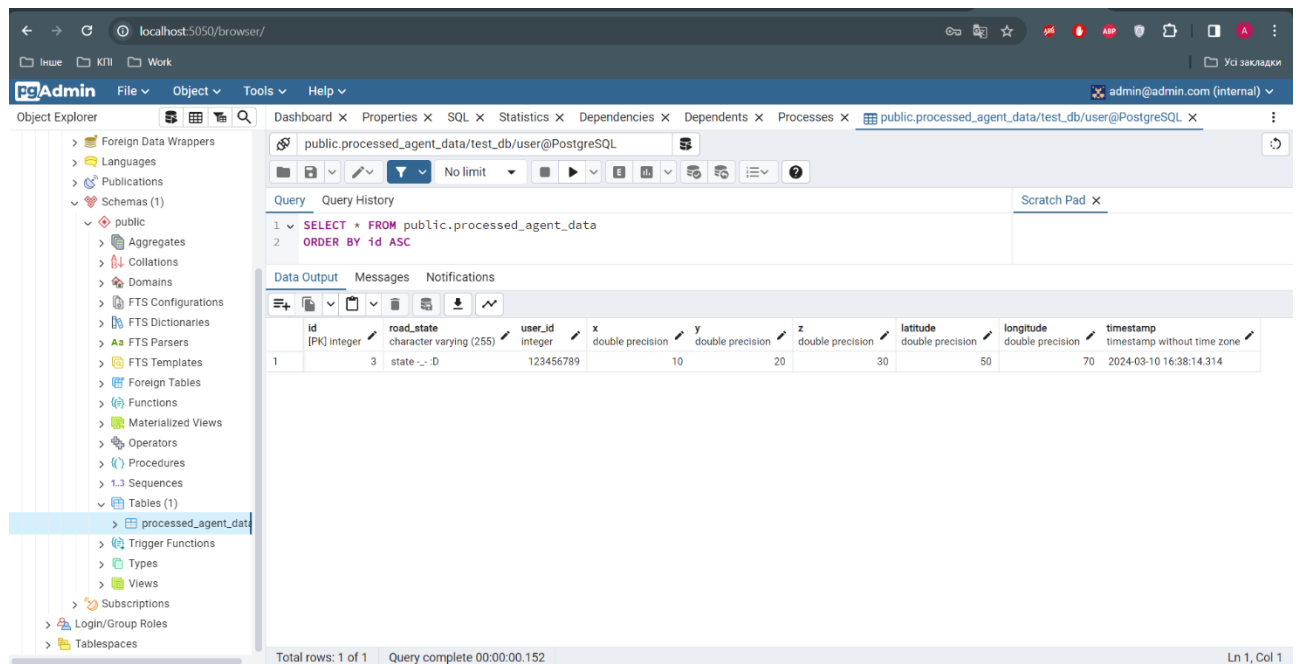
#### Response body

```
{
  "id": 2,
  "road_state": "string",
  "user_id": 111,
  "x": 0,
  "y": 0,
  "z": 0,
  "latitude": 0,
  "longitude": 0,
  "timestamp": "2024-03-10T15:08:53.776000"
}
```

#### Response headers

```
content-length: 140
content-type: application/json
date: Sun, 10 Mar 2024 16:42:03 GMT
server: uvicorn
```

pgAdmin:



**Висновок:** під час виконання даної лабораторної роботи ми ознайомились з поняттям CRUD та їх застосуваннями. З допомогою мови Python реалізували CRUD операції для endpoint-a processed\_agent\_data.