

Національний технічний університет України «КПІ ім. Ігоря Сікорського»  
Факультет Інформатики та Обчислювальної Техніки



Кафедра інформаційних систем та технологій

## Лабораторна робота №2

з дисципліни «Технології розробки вбудованих систем IoT»

Виконав:

студент групи ІС-12  
Канупа Максим

Перевірив:

Каплунов А. В.

Київ – 2025

1) Git repo: <https://github.com/Aristocrab/Labs>

2) CRUD-операції

```
# 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, [{**d,
"timestamp": d["timestamp"].isoformat()} for d in
query_values])

@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) Swagger та PgAdmin

POST:

Swagger:

The image shows the Swagger UI for a POST endpoint. The top bar indicates the method is POST and the endpoint is /processed\_agent\_data/. Below this, there are tabs for Parameters and Request body. The Request body tab is active, showing a JSON schema for the request. The schema includes fields for road\_state, agent\_data (with user\_id and accelerometer), gps (with latitude and longitude), and a timestamp. At the bottom, there is an Execute button.

```
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

The image shows a curl command and the resulting response details. The curl command is used to send a POST request to the endpoint. The response details show a 200 status code and a null response body.

```
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	<p>Response body</p> <p>null</p> <p>Response headers</p> <pre>content-length: 4 content-type: application/json date: Sun, 10 Mar 2024 16:29:10 GMT server: uvicorn</pre>

Download

Результат у pgAdmin:

localhost:5050/browser/

admin@admin.com (internal)

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

public.processed\_agent\_data/test\_db/user@PostgreSQL

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	Successful Response	No links

Media type: application/json

Controls Accept header.

Example Value | Schema

```
[
  {
    "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"
  }
]
```

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

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"
  },
  {
    "id": 3,
    "road_state": "state _- :D",
    "user_id": 123456789,
    "x": 1,
    "y": 2,
    "z": 3,
    "latitude": 0,
    "longitude": 0,
    "timestamp": "2024-03-10T16:28:04.112000"
  }
]
```

Response headers

```
content-length: 295
content-type: application/json
date: Sun, 10 Mar 2024 16:32:03 GMT
server: uvicorn
```

## PgAdmin:

localhost:5050/browser/

admin@admin.com (internal)

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

Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x public.processed\_agent\_data/test\_db/user@PostgreSQL x

Query Query History

1 SELECT \* FROM public.processed\_agent\_data

2 ORDER BY id ASC

Scratch Pad x

Data Output Messages Notifications

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	111	0	0	0	0	0	2024-03-10 15:08:53.776
2	3	123456789	1	2	3	0	0	2024-03-10 16:28:04.112

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

## GET:ID

## Swagger:

GET /processed\_agent\_data/{processed\_agent\_data\_id} Read Processed Agent Data

Parameters

Cancel

Name	Description
processed_agent_data_id * required integer (path)	<input type="text" value="3"/>

Servers

These operation-level options override the global server options.

/

Execute

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/processed_agent_data/3' \
  -H 'accept: application/json'
```

Request URL

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

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": 3,   "road_state": "state _- :D",   "user_id": 123456789,   "x": 3,   "y": 2,   "z": 3,   "latitude": 0,   "longitude": 0,   "timestamp": "2024-03-10T16:28:04.112000" }</pre> <p>Response headers</p> <pre>content-length: 152 content-type: application/json date: Sun, 10 Mar 2024 16:35:13 GMT server: uvicorn</pre>

## UPDATE : Swagger :

**PUT** /processed\_agent\_data/{processed\_agent\_data\_id} Update Processed Agent Data

Parameters

Cancel Reset

Name	Description
processed_agent_data_id * 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:

The screenshot shows the pgAdmin web interface in a browser window. The left sidebar displays the database structure, with the 'public' schema selected. The main panel shows a SQL query editor with the following query:

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

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table format. The table has 10 columns: id, road\_state, user\_id, x, y, z, latitude, longitude, and timestamp. The results show two rows of data.

id	road_state	user_id	x	y	z	latitude	longitude	timestamp
1	2	string	111	0	0	0	0	2024-03-10 15:08:53.776
2	3	state _- :D	123456789	10	20	30	50	70 2024-03-10 16:38:14.314

The status bar at the bottom indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.115'.

## DELETE :

## Swagger:

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

Parameters

Cancel

Name	Description
<b>processed_agent_data_id</b> * 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"
}
```

Download

Response headers

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

## pgAdmin:

localhost:5050/browser/

admin@admin.com (internal)

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

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

Processes

public.processed\_agent\_data/test\_db/user@PostgreSQL

Query

Query History

Scratch Pad

1 SELECT \* FROM public.processed\_agent\_data

2 ORDER BY id ASC

Data Output

Messages

Notifications

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	state --D	123456789	10	20	30	50	70	2024-03-10 16:38:14.314

Total rows: 1 of 1    Query complete 00:00:00.152    Ln 1, Col 1

## **Висновок:**

Під час виконання даної лабораторної роботи проведено ознайомлення з поняттям CRUD та його застосуванням. Реалізовано CRUD-операції для endpoint-a processed\_agent\_data мовою Python.