

Session 5 Homework: FastAPI + PostgreSQL + EC2

Due: Before next session

Objective

Integrate the HR database with FastAPI, write tests, and deploy on AWS EC2.

Part 1: FastAPI Integration

Complete the TODO stubs in `bdi_api/s5/exercise.py` to create a REST API for the HR database.

Endpoints to implement

Method	Endpoint	Description
POST	<code>/api/s5/db/init</code>	Create all HR tables (run schema SQL)
POST	<code>/api/s5/db/seed</code>	Populate tables with sample data
GET	<code>/api/s5/departments/</code>	List all departments
GET	<code>/api/s5/employees/</code>	List employees with department name (paginated)
GET	<code>/api/s5/departments/{dept_id}/employees</code>	Employees in a specific department
GET	<code>/api/s5/departments/{dept_id}/stats</code>	Department KPIs: avg salary, employee count, project count
GET	<code>/api/s5/employees/{emp_id}/salary-history</code>	Salary evolution for an employee

Setup

```
cd bts-bdp-assignment
pip install -r requirements.txt
```

For local development, use Docker PostgreSQL:

```
cd ../bts-bdp-exercises/s5
docker compose up -d
```

The database URL is configured via `settings.db_url` (env variable `BDI_DB_URL`):

```
export BDI_DB_URL="postgresql://postgres:postgres@localhost:5432/hr_database"
```

You can access it in your code with `settings.db_url`.

Run the app:

```
uvicorn bdi_api.app:app --reload --port 8080
```

Visit <http://localhost:8080/docs> to test your endpoints.

Hints

- Use SQLAlchemy or psycopg2 (you already practiced both in class)
- The `hr_schema.sql` and `hr_seed_data.sql` files contain the SQL you need
- For pagination, use SQL `OFFSET` and `LIMIT`
- For stats, use `AVG()`, `COUNT()`, and `JOINS`
- Make `/db/init` and `/db/seed` **idempotent**: use `DROP TABLE IF EXISTS` before creating, so they can be called multiple times without errors (the tests call both endpoints before each test)

Part 2: Tests

Make sure all tests pass:

```
pytest tests/s5/ -v
```

The tests in `TestItCanBeEvaluated` verify your endpoints return the expected JSON structure.

Part 3: Deploy on EC2

1. Launch an EC2 instance (Amazon Linux 2 or Ubuntu, `t2.micro`)
2. Install Python 3.12+ and pip
3. Clone your repository and install dependencies
4. Configure `BDI_DB_URL` to point to your RDS instance
5. Run:

```
uvicorn bdi_api.app:app --host 0.0.0.0 --port 8080
```

6. Configure the security group to allow inbound traffic on port 8080

Deliverables (submit on Canvas)

1. **Code**: Link to your GitHub repository with completed `bdi_api/s5/exercise.py`
2. **Tests**: Screenshot of `pytest tests/s5/ -v` passing
3. **EC2**: Screenshot of your API docs page (`/docs`) running on EC2 public IP
4. **Architecture**: Brief description of your deployment (which database, how it connects)