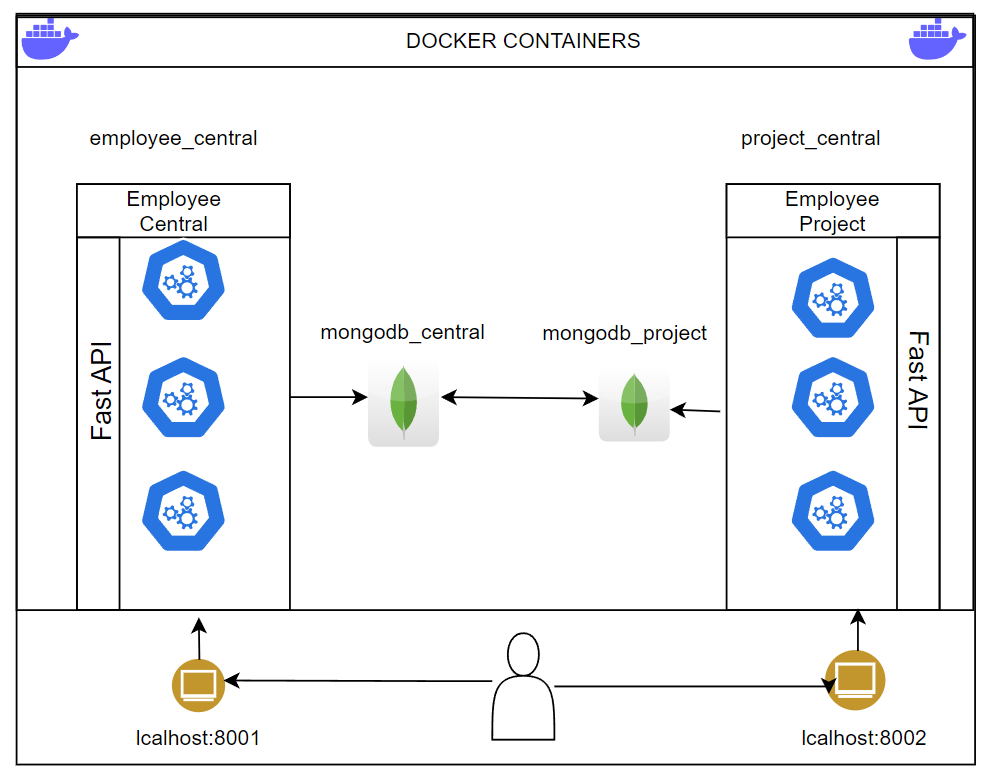
**Microservices for Employee and Project Management with MongoDB, Docker & Kubernatess**

This assignment focuses on developing two microservices, **EmployeeCentral** and **ProjectCentral**, each managing different aspects of employee and project data. Both services are deployed using Docker containers and communicate with each other to maintain consistency between their respective MongoDB databases. The goal is to demonstrate how microservices architecture can be used for efficient, scalable management of different service domains while ensuring seamless data communication between them.

ARCHITECTURE DIGRAM:



**Overview**

**a). EmployeeCentral:**

**Purpose**: Handles operations related to employees such as adding new joiners, updating employee details, handling resignations, and fetching employee information.

* **Endpoints**:
  + POST /employee/new\_joiner: Add a new employee to the system.
  + PUT /employee/{employee\_id}: Update an employee's details (e.g., position, salary, department).
  + DELETE /employee/{employee\_id}/resignation: Handle employee resignation by deleting the employee.
  + GET /employees: Get a list of all employees.
  + GET /employee/{employee\_id}: Get employee details by ID.

**b). ProjectCentral:**

**Purpose**: Manages projects and promotions, communicating with the EmployeeCentral service to fetch and update employee data and managing project assignments within its own database.

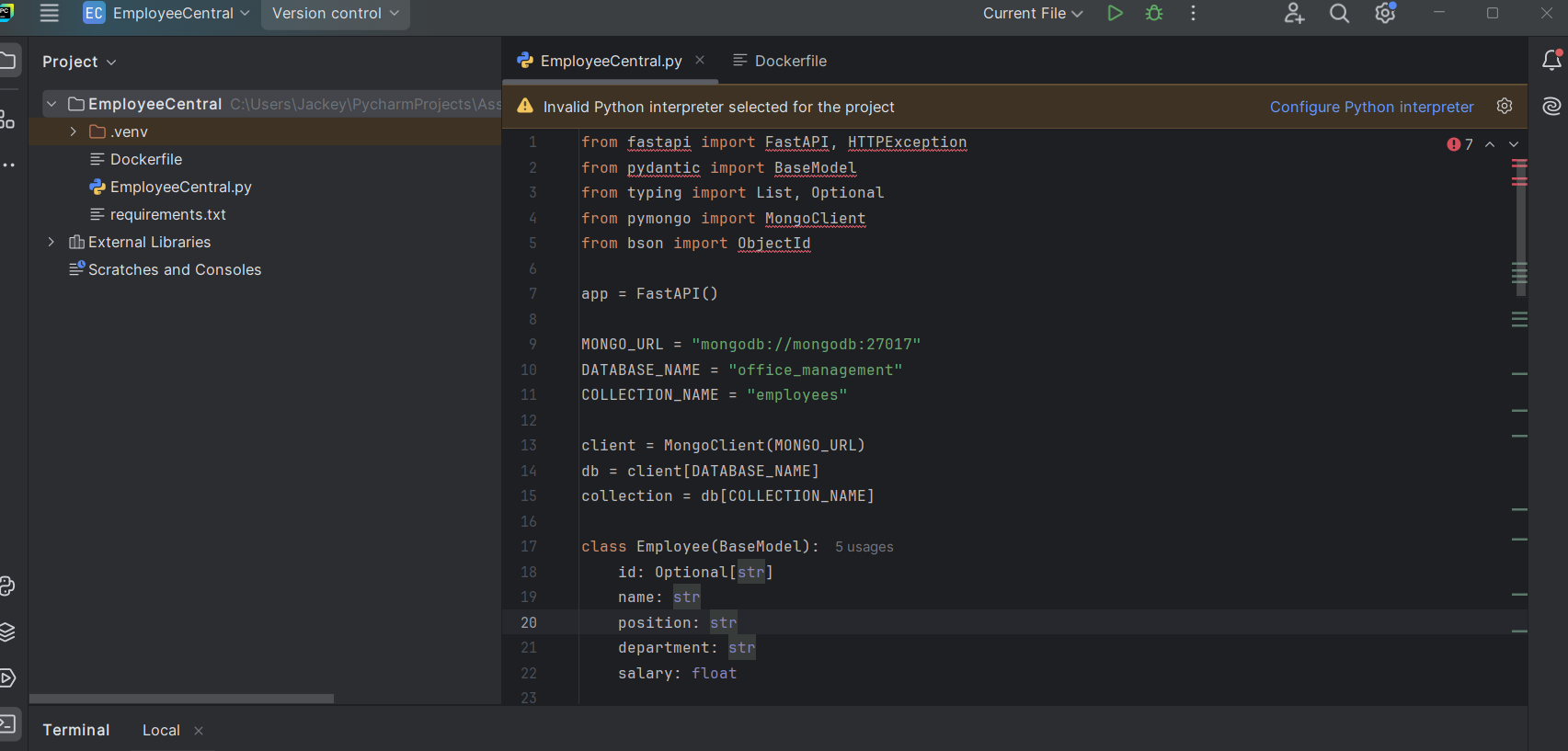
* **Endpoints**:
  + PUT /employee/{employee\_id}/promotion: Promote an employee and update their position and salary both in **EmployeeCentral** and **ProjectCentral**.
  + POST /employee/{employee\_id}/project: Add a project to an employee's record, fetching employee data from **EmployeeCentral** and saving the project details in **ProjectCentral**'s database.
  + GET /employee/{employee\_id}/projects: Get the list of projects assigned to an employee, stored in **ProjectCentral**'s MongoDB.

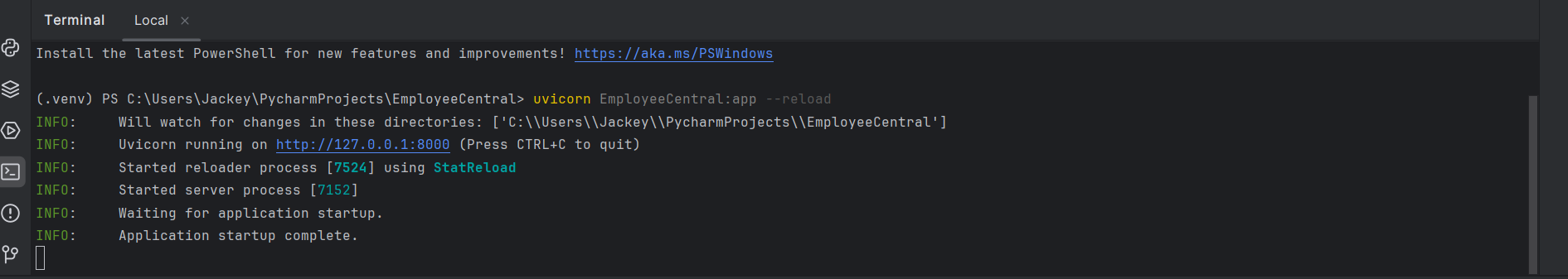
**c). MongoDB Integration:**

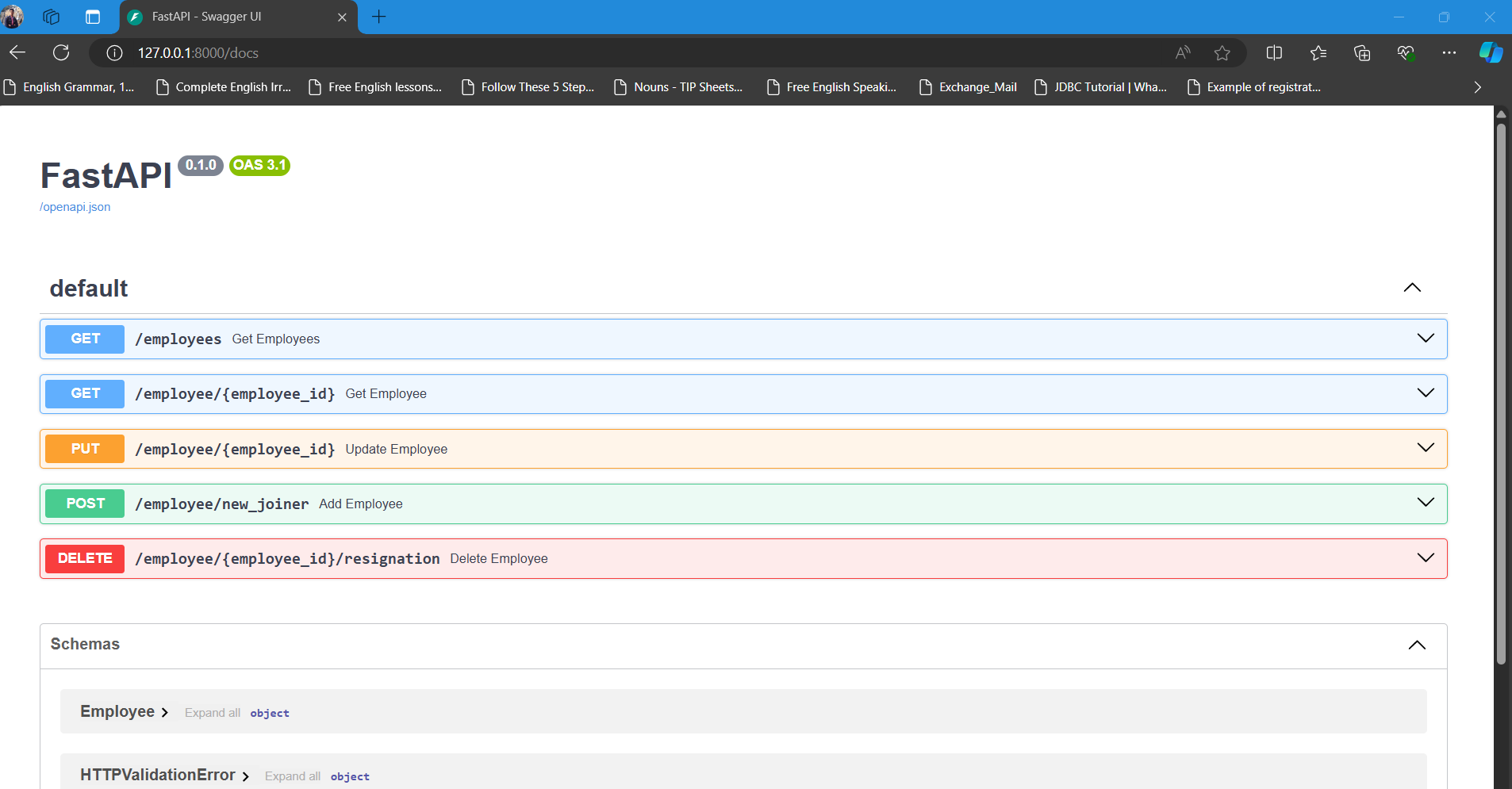
Each microservice has its own MongoDB instance:

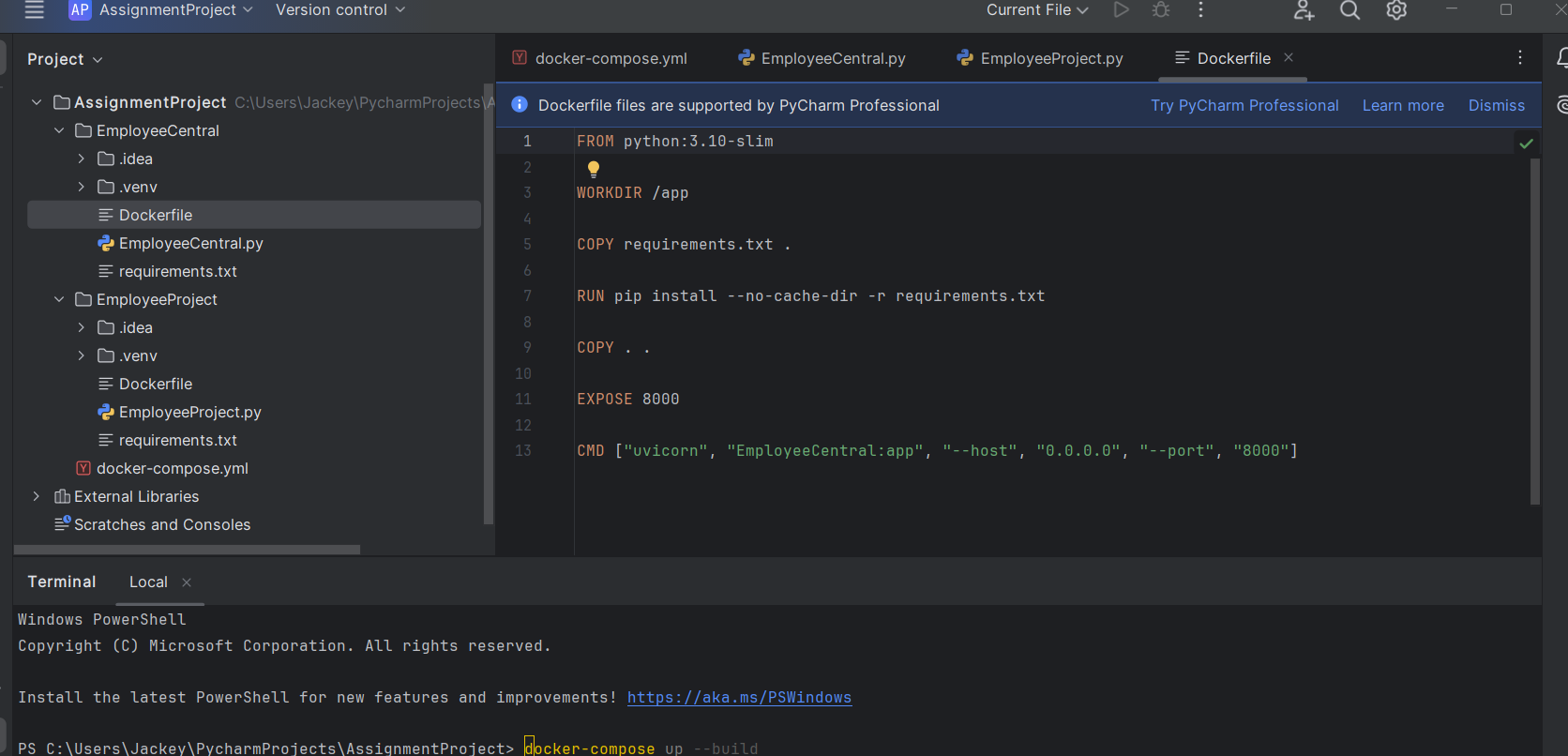
* **EmployeeCentral MongoDB**:
  + Manages employee data such as name, position, department, and salary.
* **ProjectCentral MongoDB**:
  + Manages project data associated with employees, storing projects each employee is assigned to.

Both microservices interact with their respective databases but communicate with each other to ensure the necessary data is updated consistently. For example, when an employee is promoted via **ProjectCentral**, their promotion data is updated in **EmployeeCentral** to maintain data integrity.

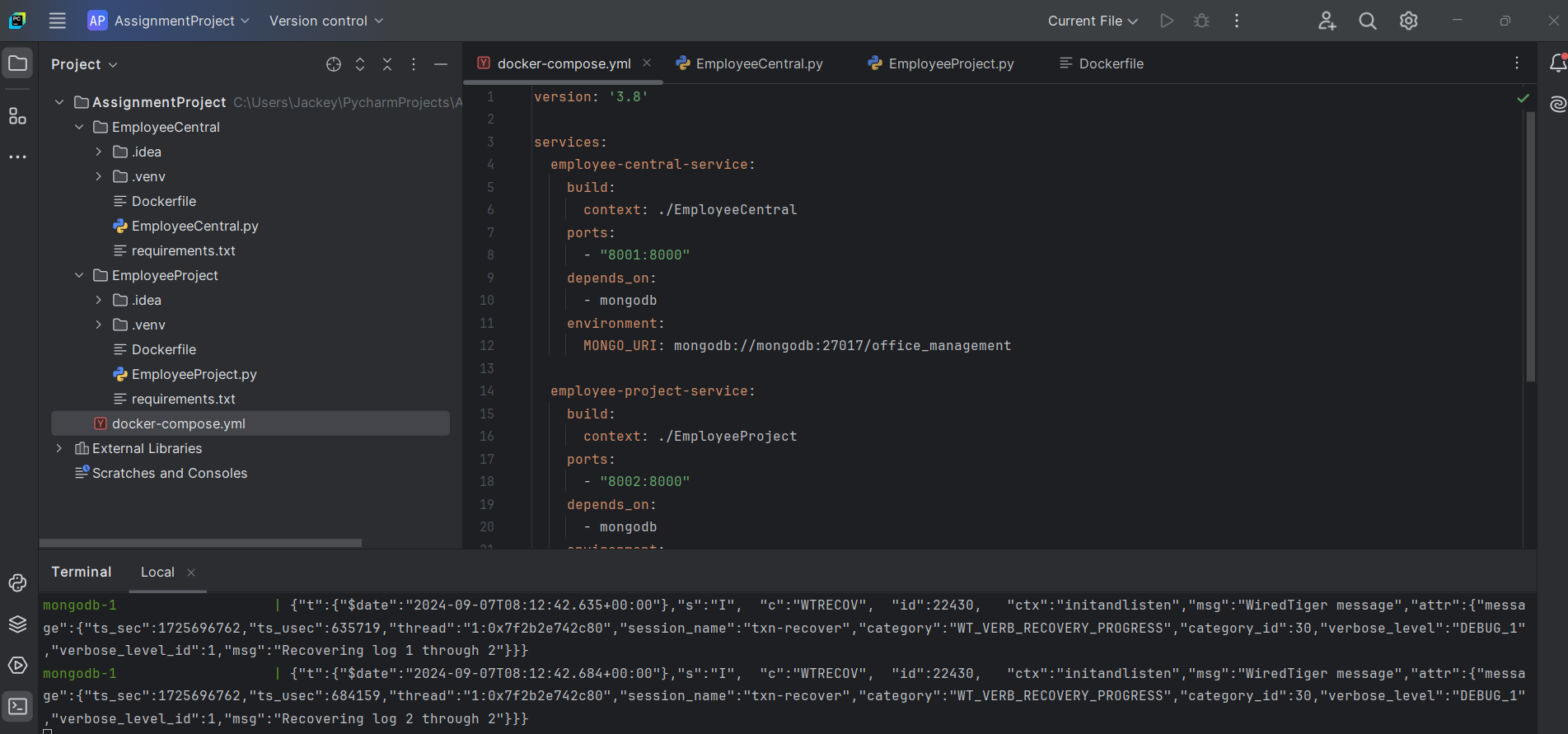




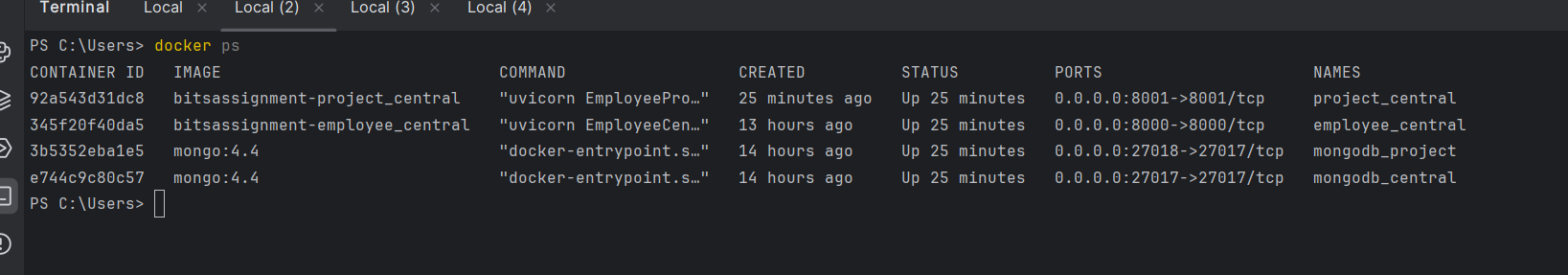


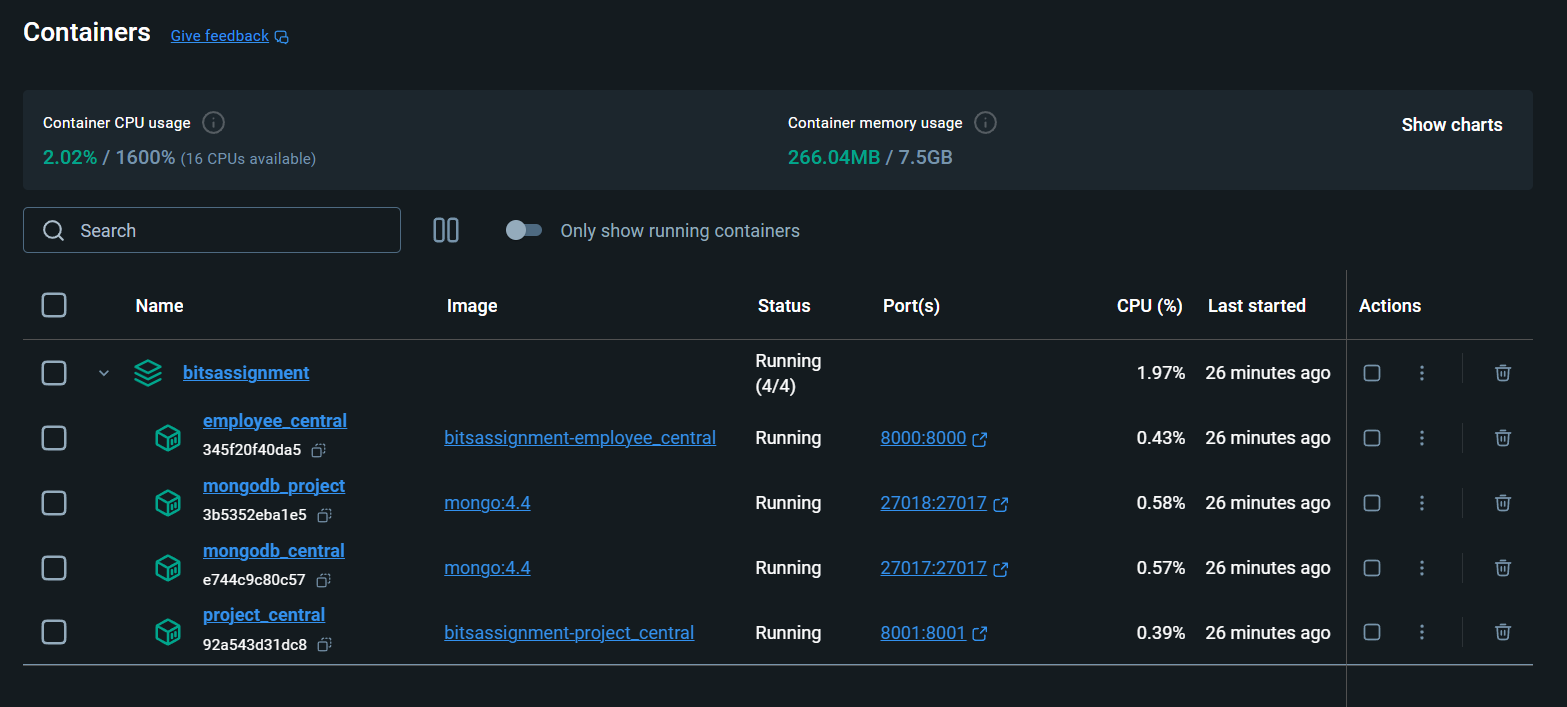


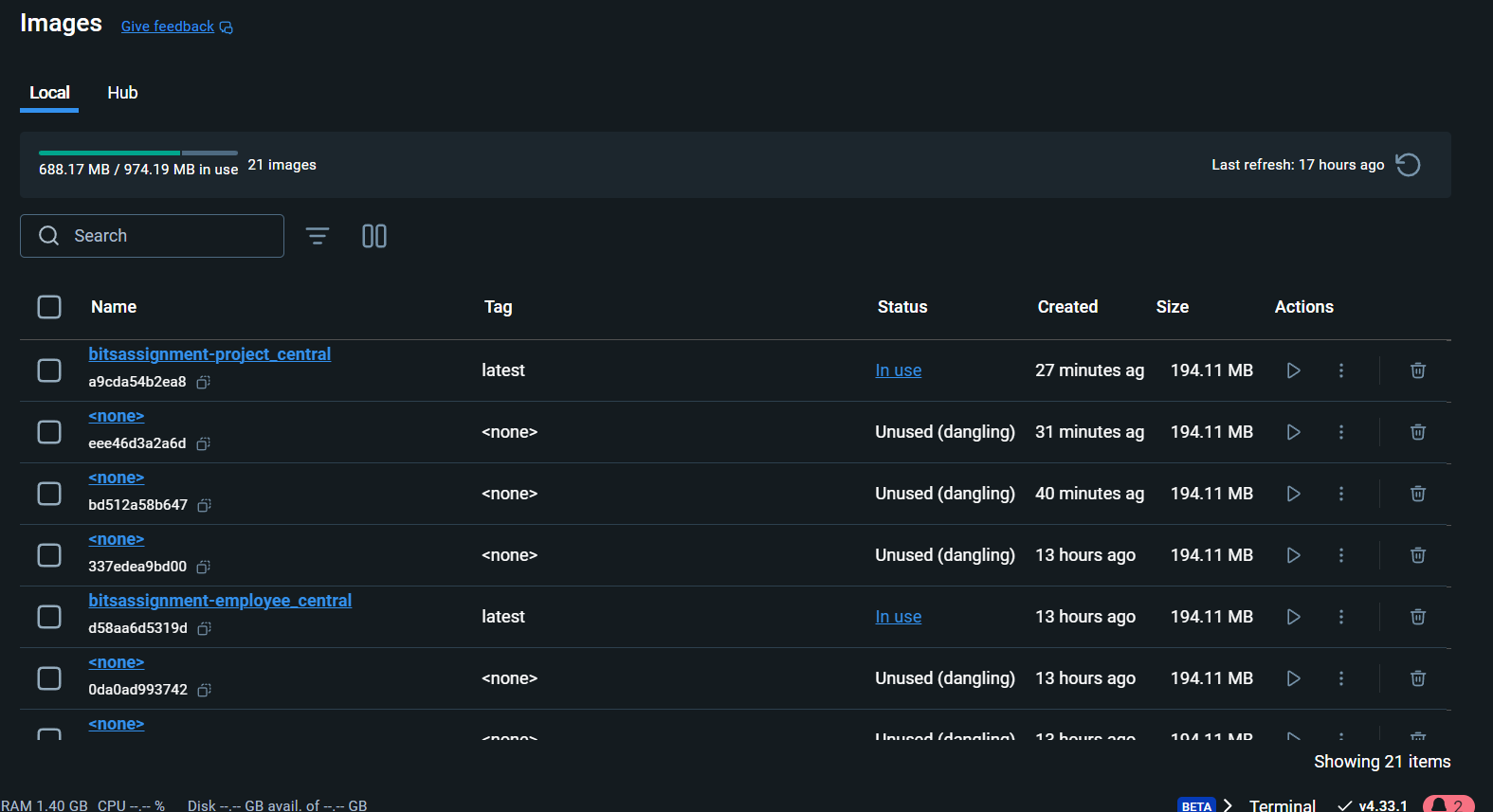
docker-compose up --build



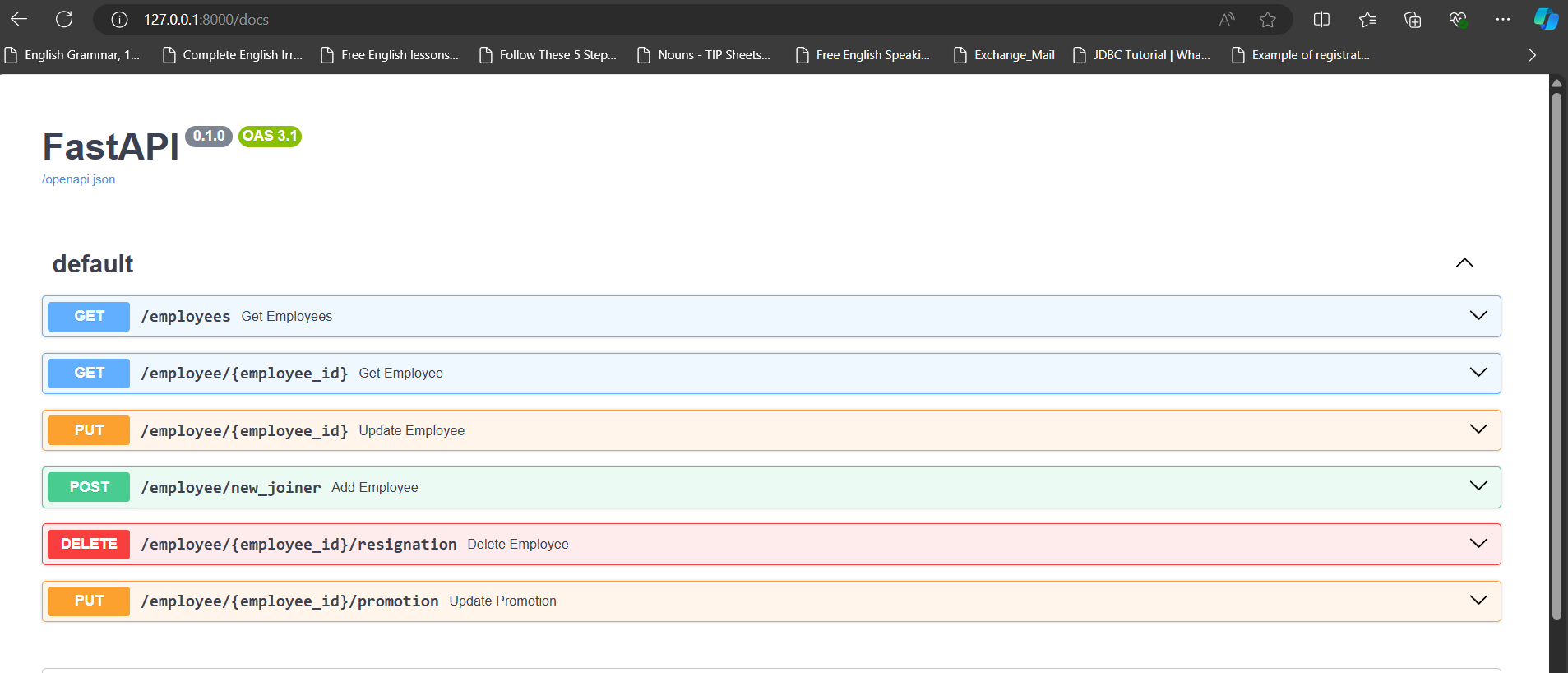
Docker ps



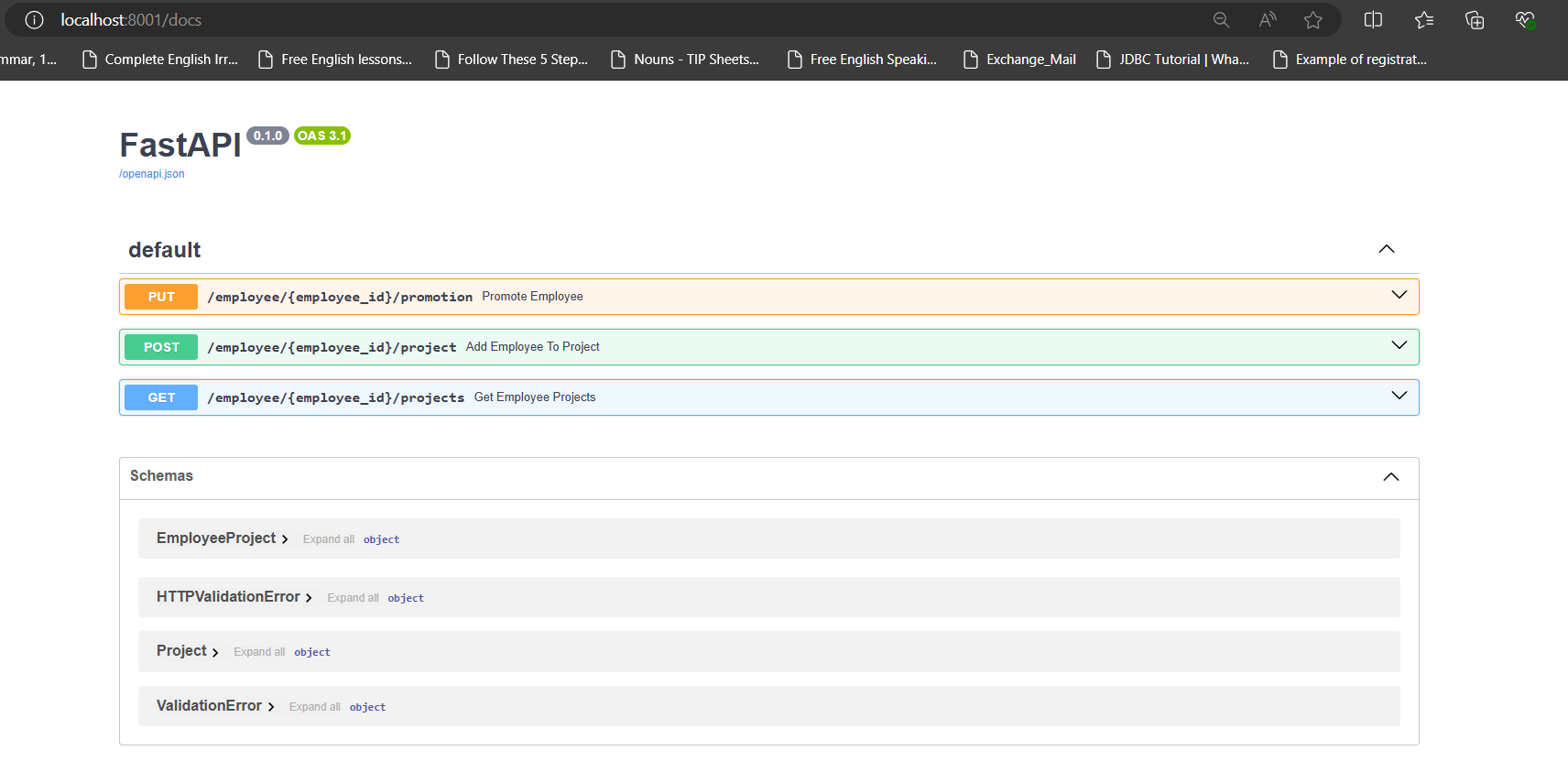


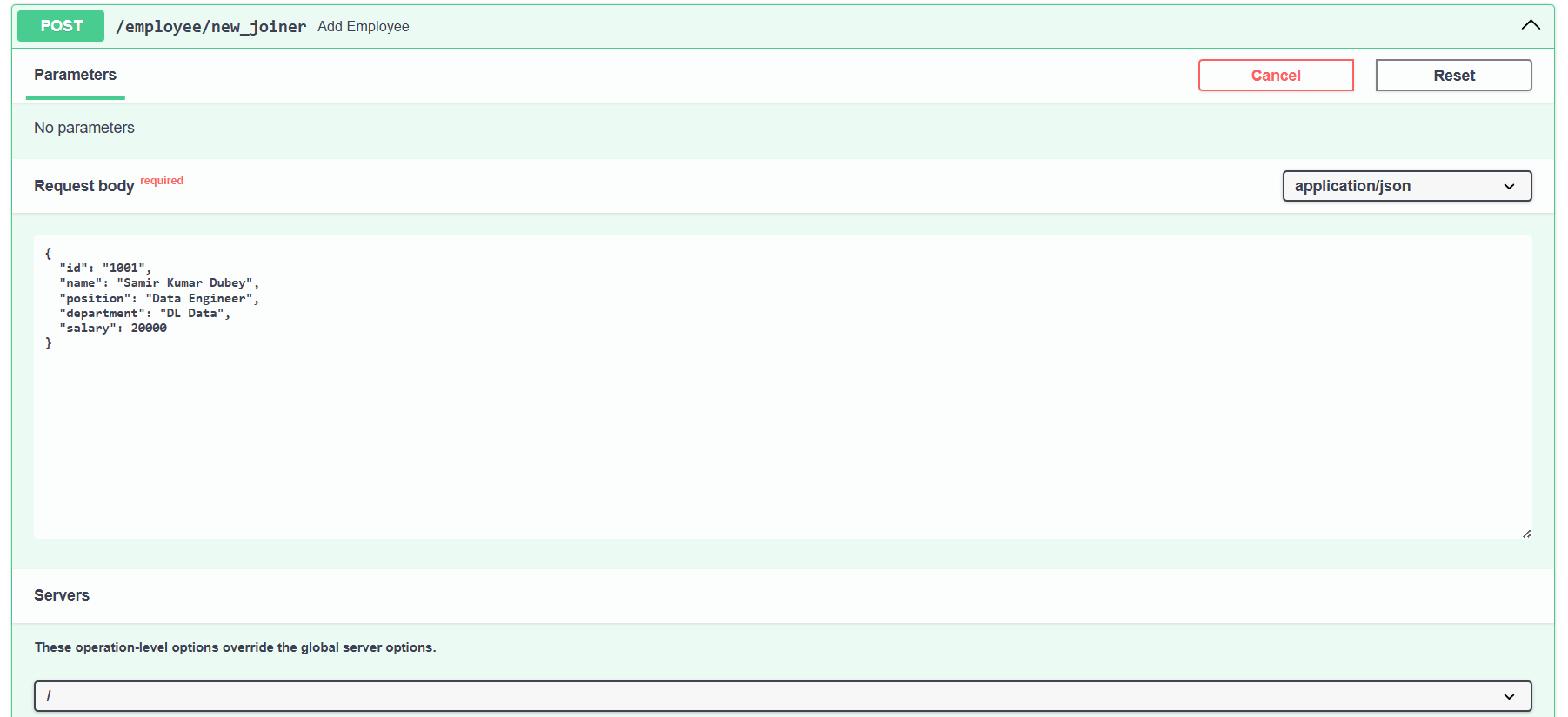


http://127.0.0.1:8000/doc

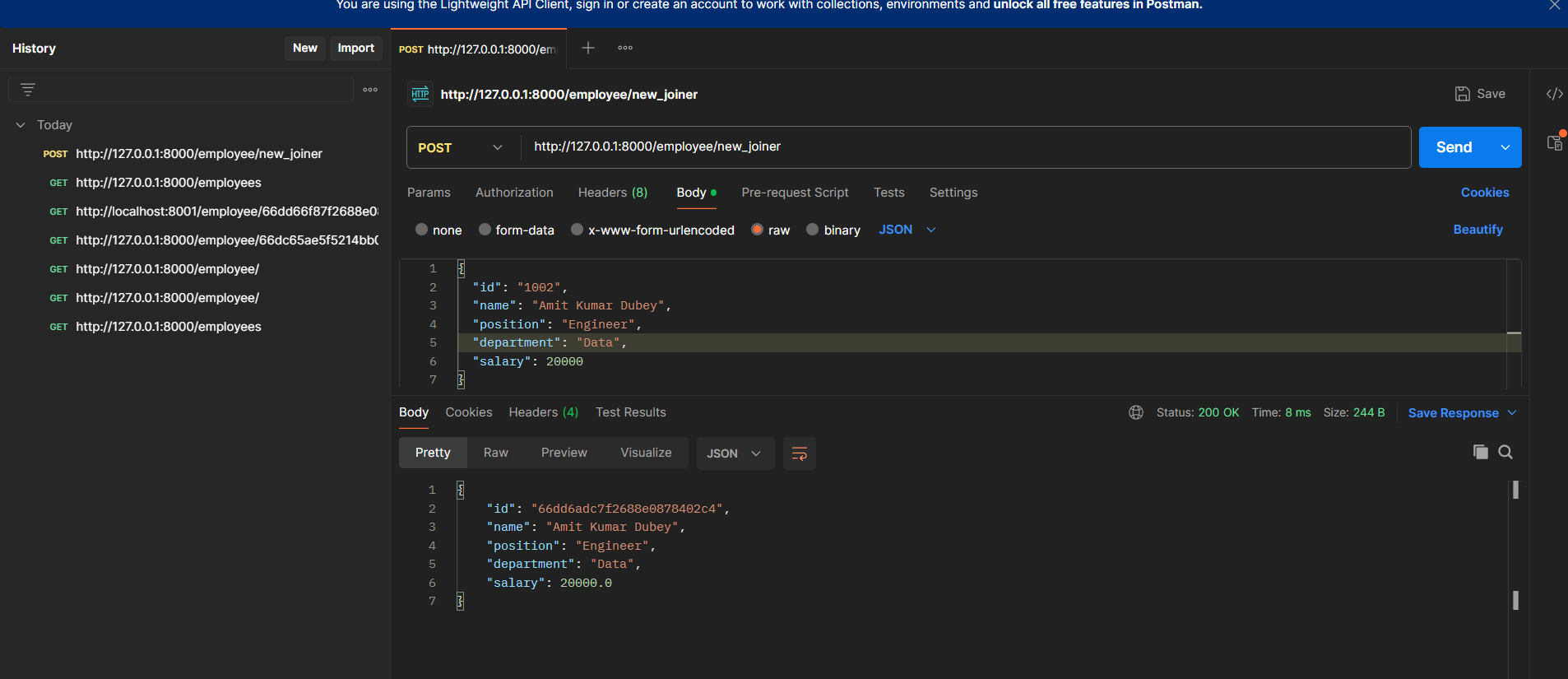


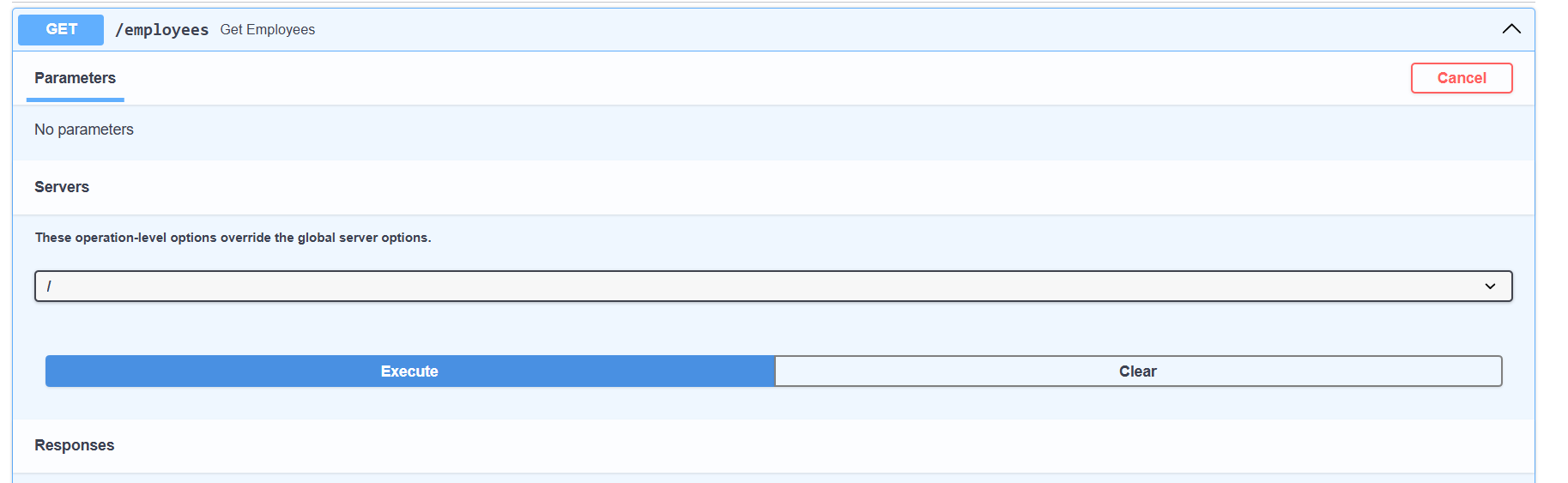
http://127.0.0.1:8001/doc

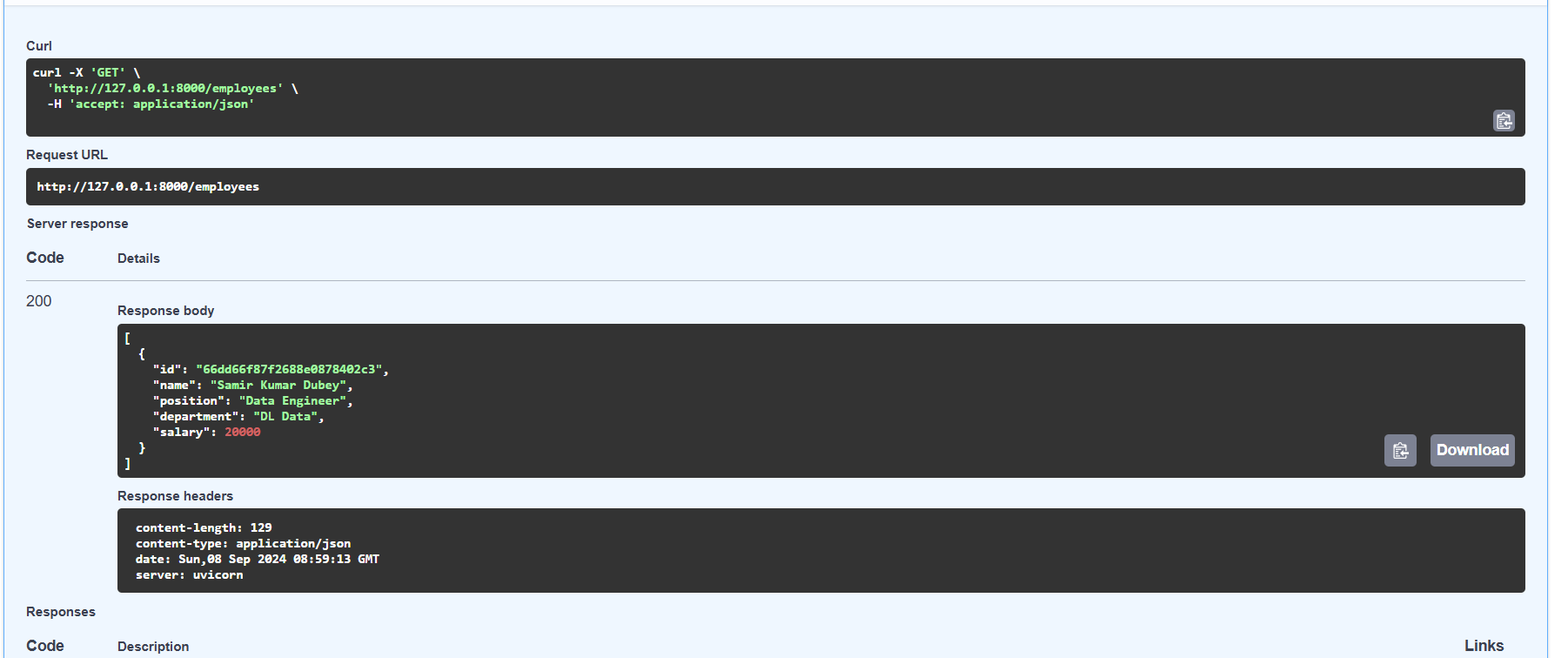


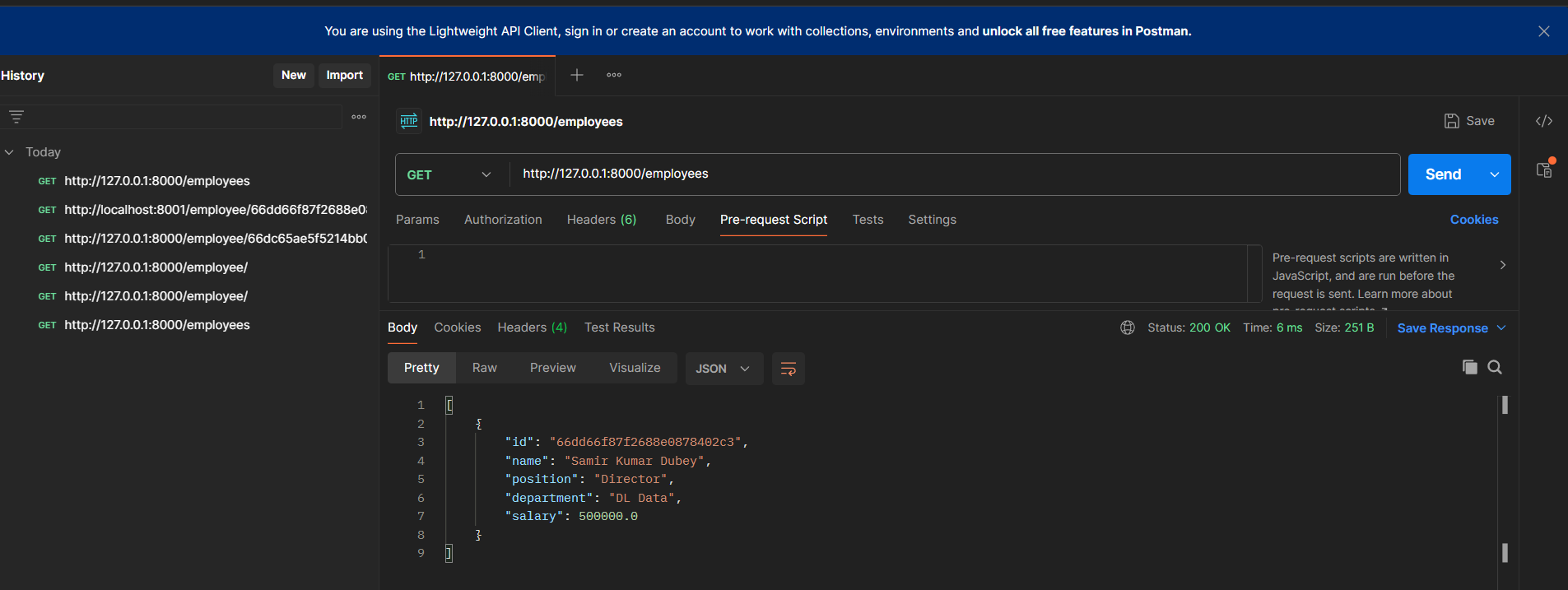


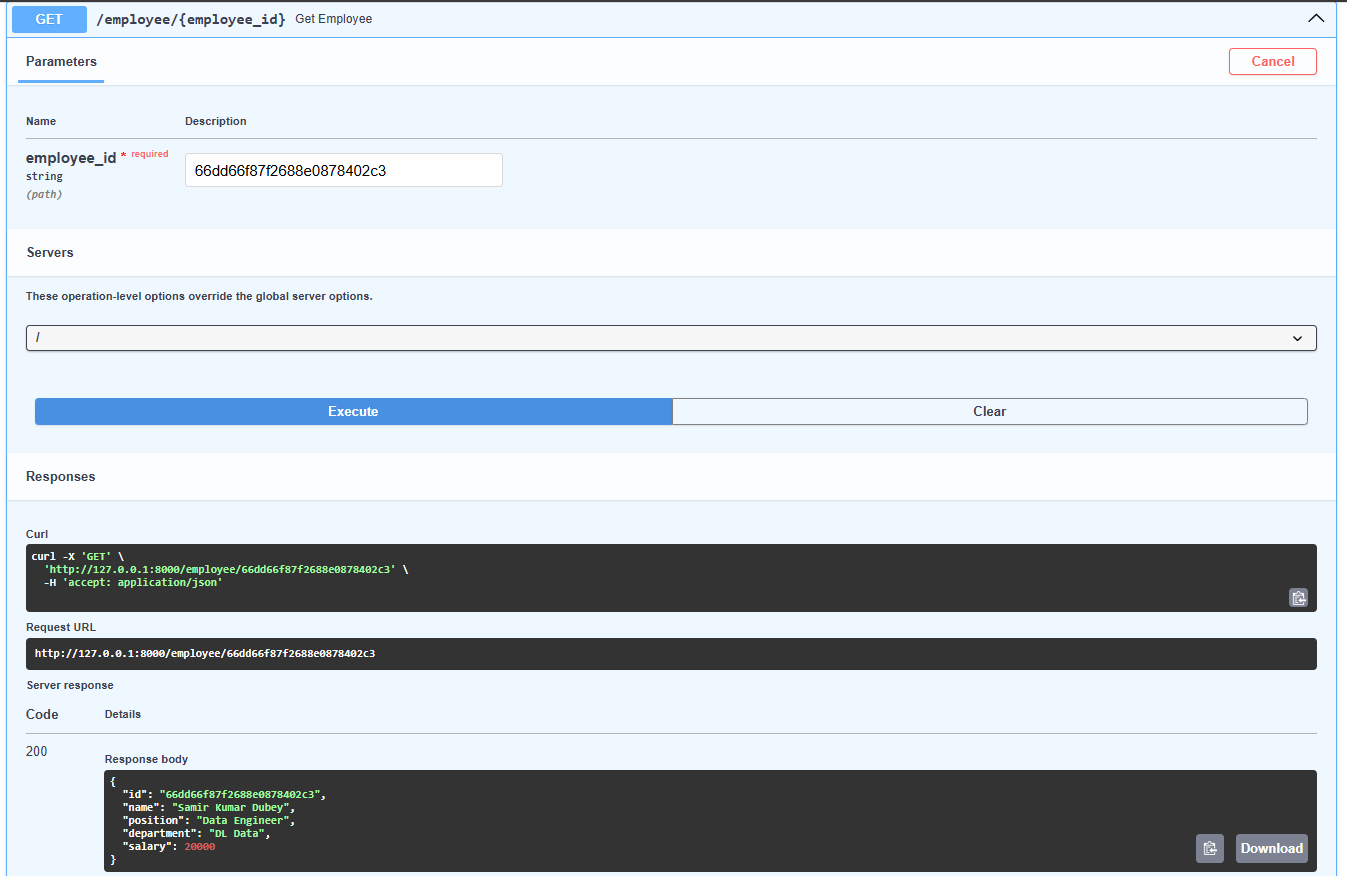


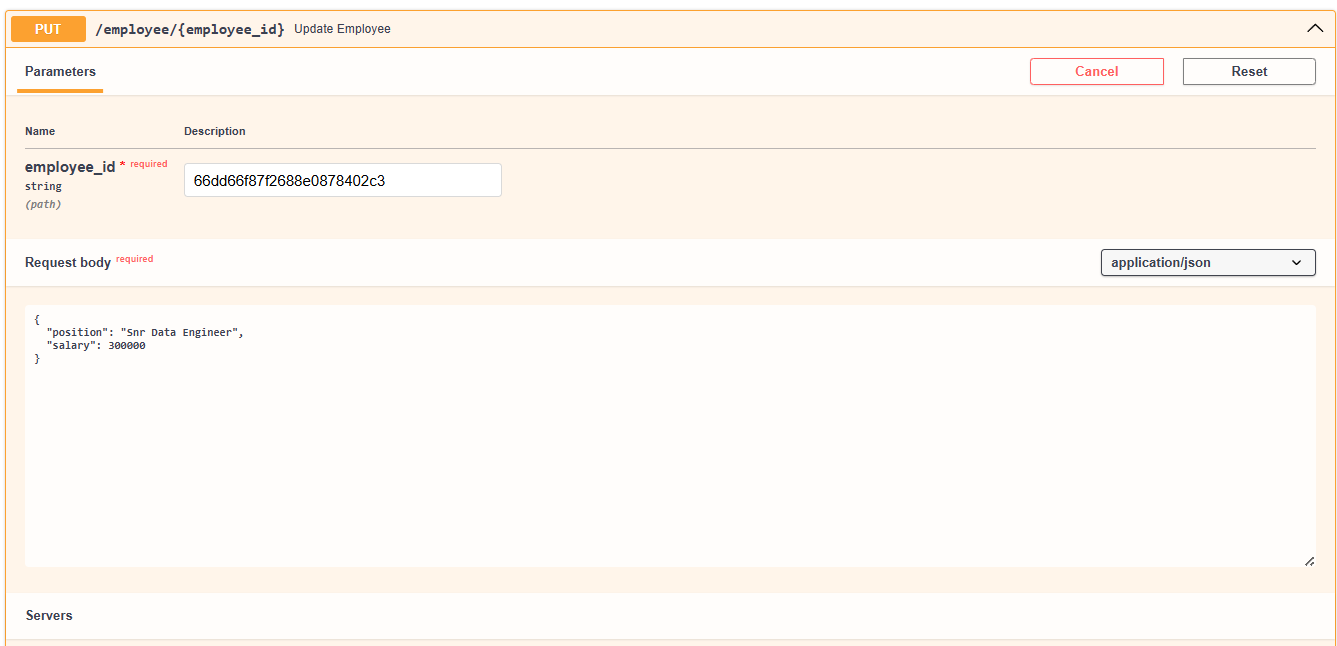




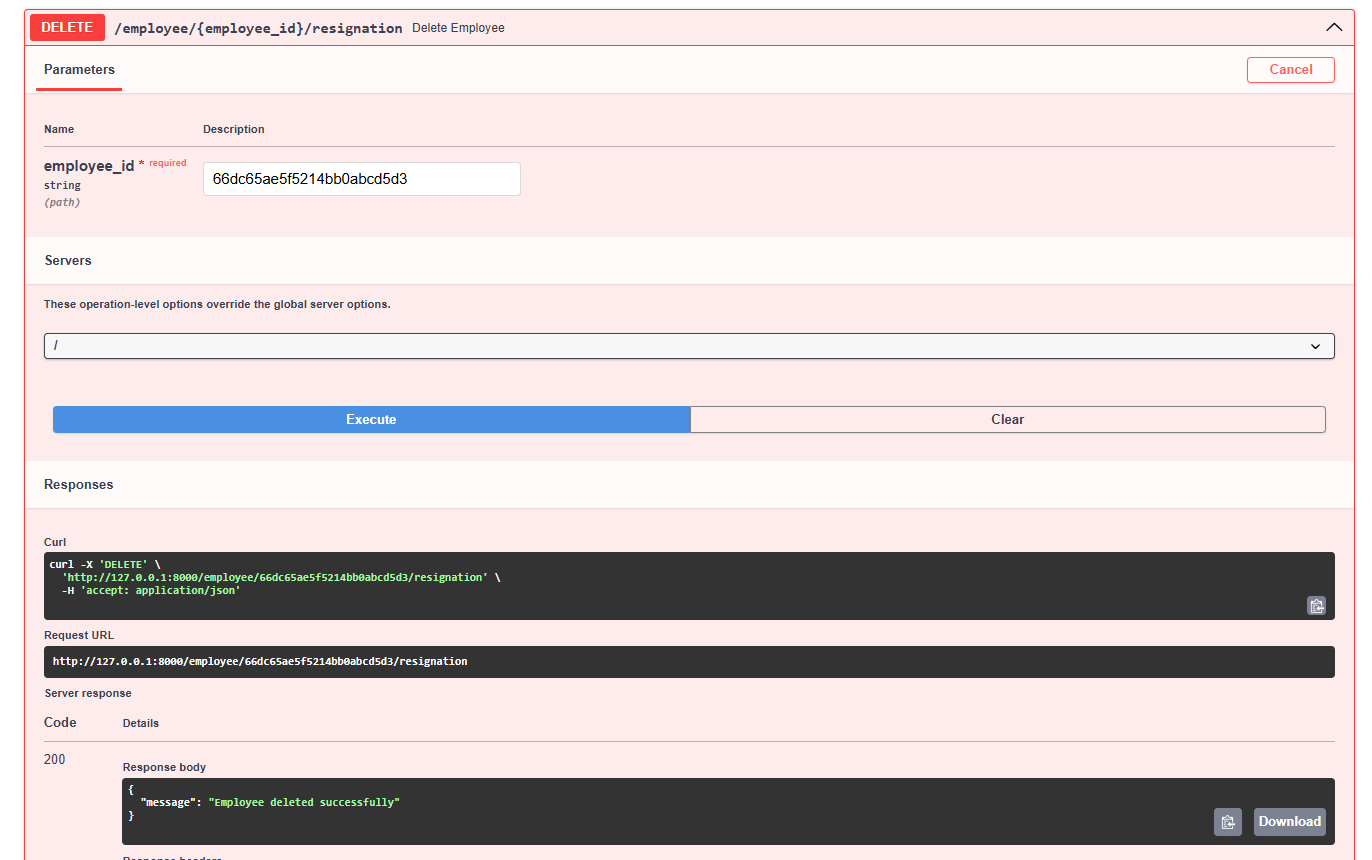












2nd Micreoservice

