

2. Create the following two database in in-memory database:

DNO	DNAME
10	Admin
20	Accounts
30	Sales
40	Marketing
50	Purchasing

ENO	ENAME	DNO	SALARY
1	Amal	10	30000
2	Shyamal	30	50000
3	Kamal	40	10000
4	Nirmal	50	60000
5	Bimal	20	40000
6	Parimal	10	20000

Design the two APIs

Important Points:

- I) Run the server on port 9000. The APIs should only serve on port 9000.
- II) Exceptions should be handled.
- III) Casting if any, should be type safe.
- IV) You need to explain your answer.
- V) Keep your results ready in a HTTP client application like postman.

a. Write an Api to return response depending on ENO.

E.g.: `http://localhost:9000/api?ENO=1`

This Api should return the details of the employee whose ENO is 1.

Step 1:

These are the package and import statements, similar to the previous code.

Step 2:

This class is the main class of the program and defines the static maps for storing department and employee information.

Step 3:

The main method populates the department and employee databases, creates an HTTP server listening on port 9000, associates a `DepartmentHandler` instance with the `/api` context, and starts the server.

Step 4:

This class defines the structure of an `Employee`. It has properties for employee ID, name, department number (dno), and salary. It also has getter methods to access these properties.

Department Class:

Step 5:

This class defines the structure of a department, including its number (dno) and department name (dname). It provides getter methods for accessing these properties.

Step 6:

This inner static class implements the `HttpHandler` interface, which handles incoming HTTP requests. The `handle` method processes the request by extracting the `ENO` from the query parameter and looking up the corresponding employee's details. If found, it creates a `Department` instance and sends the employee's details along with the department name as a response.

Execution process:

The program creates an HTTP server on port 9000 that listens for requests on the `/api` endpoint. When a client sends a request with an `ENO` query parameter, the server looks up the corresponding employee's details and department name. If the employee is found, the server responds with the employee's details and department name. If the employee is not found or the request is invalid, appropriate error responses are sent.