Proposal: Employee Management RESTful Web Service

Overview

This proposal outlines the design of a RESTful web service to manage a list of employees. The service will be implemented using Java Spring Boot and hosted on the HPE GreenLake private cloud platform. It will support GET, POST, DELETE, and UPDATE operations and store employee data securely in the cloud.

Tools and Technologies

1. Java Spring Boot

- A robust framework for building RESTful APIs.
- Provides tools for rapid development and built-in support for handling HTTP requests.

2. HPE GreenLake Cloud Platform

- Offers scalable private cloud infrastructure.
- Provides services for secure data storage and management.

3. Database

- A relational database such as MySQL or PostgreSQL will be deployed on GreenLake to store employee data persistently.

API Design

1. Data Model

- Employee entity with the following fields:
- first_name (String): Employee’s first name.
- last_name (String): Employee’s last name.
- employee_id (Integer): Unique identifier.
- email (String): Employee’s email address.
- title (String): Job title.
- The Employee class will include private variables for each field, along with getter and setter methods.

2. Endpoints

1. GET /employees: Retrieve a list of all employees.

- Response: JSON array of employee objects in the format: { "Employees": [{ "employee_id": "string", "first_name": "string", "last_name": "string", "email": "string", "title": "string" }] }

2. POST /employees: Add a new employee.

- Request Body: JSON object with all fields in the format: {   "employee_id": "string",   "first_name": "string",   "last_name": "string",   "email": "string",   "title": "string" }
- Response: Success message with the created employee’s details.

3. PUT /employees/{employee_id}: Update an employee’s details.

- Request Body: JSON object with updated fields in the same format as POST
- Response: Success message with the updated employee’s details.

4. DELETE /employees/{employee_id}: Remove an employee by their ID.

- Response: Success message.

3. Controller Class

- A Controller class will handle GET, POST, PUT, and DELETE operations to manage entries in the database.

4. Error Handling

- 400 Bad Request: Invalid input data.
- 404 Not Found: Employee not found.
- 500 Internal Server Error: Server-side issues.

Cloud Infrastructure

- Deployment: The application will be deployed on HPE GreenLake.
- Security: API requests will require authentication to ensure secure access.

Conclusion

This RESTful web service will provide an efficient way to manage employee records while leveraging Java Spring Boot’s capabilities and HPE GreenLake’s secure cloud infrastructure. The design ensures compliance with the specified requirements.