To create this service, we'll start by developing an Employee class in Java Spring Boot. This class should encompass private fields for various attributes, such as first_name, last_name, employee_id, email, and title. Additionally, it should feature corresponding setter and getter methods for each of these attributes. Subsequently, we'll craft an Employee management class to handle the complete employee list.

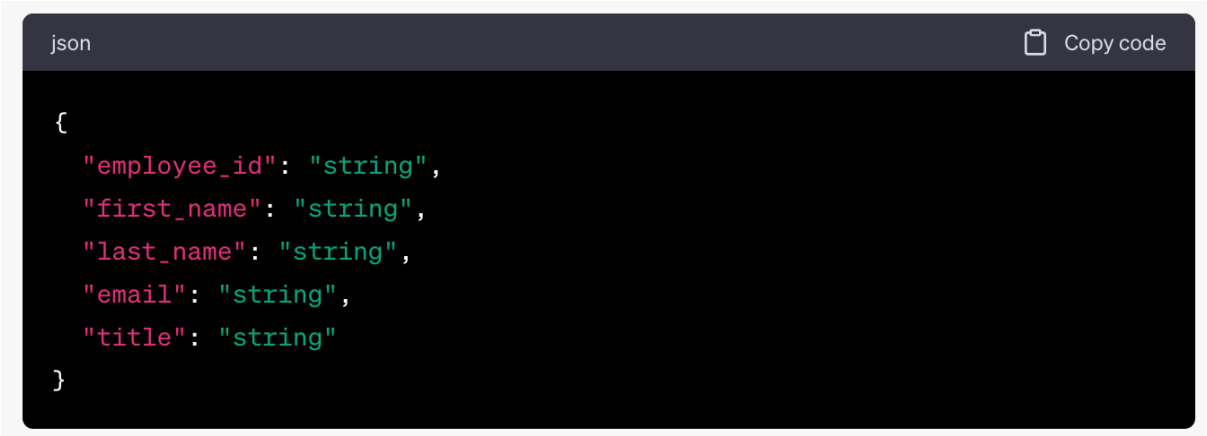
Following that, we'll establish a Controller class responsible for overseeing interactions with the database through GET, POST, DELETE and PUT commands. Users can employ the GET command to retrieve the entire user data list, utilize the POST command to insert an individual user, deploy the DELETE command to eliminate an entry, and use the PUT command to update an existing record.

To retrieve the comprehensive list of employees, a GET request can be dispatched to the service, and the response will be in the following format:

json Copy code

```
{
  "Employees": [
    {
      "employee_id": "string",
      "first_name": "string",
      "last_name": "string",
      "email": "string",
      "title": "string"
    },
    {
      "employee_id": "string",
      "first_name": "string",
      "last_name": "string",
      "email": "string",
      "title": "string"
    },
    {
      "employee_id": "string",
      "first_name": "string",
      "last_name": "string",
      "email": "string",
      "title": "string"
    }
  ]
}
```

For individual employees, we can utilize the POST command to add a new employee, the DELETE command to remove an employee, and the PUT command to modify employee details. These commands should follow this structure:



Lastly, we'll require a hosting solution for this data, which is where the GreenLake Cloud Platform becomes valuable. This platform enables us to privately host our data on-site, leveraging the architecture provided by GreenLake. They offer cloud-native infrastructure, allowing companies to deploy resources locally at a cost-effective rate.