To build this service, we’ll first need to create an Employee class in Java Spring Boot. This should include private variables for each field, including first_name, last_name, employee_id, email, and title. Additionally, it should include setter and getter methods for each of these variables. Then, we’ll need to create an Employee class that manages the full list of employees.

Next, we’ll need to create a Controller class. This is the class responsible for controlling the entries to the database using the GET, POST, DELETE, and PUT commands. Users should be able to use the GET command to acquire the full list of user data, while the POST command should be used to add an individual user. Additionally, the DELETE command can be used to delete an entry, and the PUT command. Users should be able to use the get command to acquire the full list of user data, while the post command should be used to add an individual user. Additionally, the delete command can be used to delete an entry, and the put command can be used to update an existing entry.

To get the full list of employees, we can send a GET request to the service and receive a response in the format:

```
{
    “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 send the POST command to add a new employee, send the DELETE command to delete an employee, and send the PUSH command to update an employee. These commands can be structured as follows:

```
{
    “employee_id”: “string”,
    “first_name”: “string”,
    “last_name”: “string”,
    “email”: “string”,
    “title”: “string”
}
```

Finally, we’ll need to host this data somewhere. That’s where the GreenLake Cloud Platform will come into use. With that platform, we can host our data privately on the site using architecture that GreenLake provides. They deliver cloud-native infrastructure that makes it possible for companies to deploy resources at a low cost locally.