Business Requirements

The HR department in CompanyXYZ needs to build a system for storing and querying employee data.

The data model they described is as follows:

- Each employee record has a "last name" and a "first name". The maximum number of characters allowed for those fields is 32.
- Each employee has a date of birth.
- Each employee has a direct manager (who is also an employee in the company).
- Each employee has "salary" (a decimal representation) and "department" (string representation) data associated.

The system need to contain the following APIs:

- An API to store an employee data;
- An API to retrieve the employee data;
- An API to update employee data;
- An API to remove an employee from the system;
- An API to return the following reports:
  - Obtain the employee who has the biggest salary in a given department;
  - Obtain the manager who has the most "direct" employees coordinated by him.

Bonus Points:

- Add another API to bulk upload employees (file upload with a csv, or json containing thousands of employees)
- Add paging to the employee retrieval API
- Obtain the top `n` best paid employees in a given department
- Return another report: the management tree, from the top CEO down to the lowest employees

Technical Requirements

The system is imagined as a micro-service written in Spring Boot.
The persistence should be done using a No-SQL database. Preferably Elasticsearch or MongoDB.
Write automated tests for the micro-service. It's up to you to decide what kind of tests you want to write.
The project should use gradle as a build tool.
(Good to have) Dockerization.

Also, the solution should be uploaded to some public git repository (GitHub, etc.)