**EBF**

***Employee Storage Service***

Confidentiality and Copyright

Proprietary, confidential data. All rights reserved.

Distribution and/or reproduction of these documents or any of their parts, their utilisation or communication of their contents to third parties is not allowed, unless formally conceded. Contravention will cause damages.

Copyright © 2019 by S4 Integration (Pty) Ltd.

S4 Integration (Pty) Ltd.

1 Leadwood Crescent

Fairview

Port Elizabeth

South Africa

6065

[www.s4.co.za](http://www.s4.co.za/)

**Table of Contents**

[Confidentiality and Copyright 2](#__RefHeading___Toc1260_4150984532)

[1 System Overview 4](#__RefHeading___Toc1262_4150984532)

[Data definition 4](#__RefHeading___Toc1264_4150984532)

[Delivery 4](#__RefHeading___Toc1266_4150984532)

[2 Test Cases 5](#__RefHeading___Toc1268_4150984532)

[2.1 Feature 5](#__RefHeading___Toc1270_4150984532)

[2.2 Employees 5](#__RefHeading___Toc1272_4150984532)

[2.3 Feature 6](#__RefHeading___Toc1274_4150984532)

[2.4 Feature 7](#__RefHeading___Toc1276_4150984532)

# System Overview

We're building an employee storage service for companies.

A company admin needs to be able to:

* create employee records for company,
* view them in a list and detail view for company,
* update the employee records and delete them for company,
* find the average salary for the company

Your task is to build the backend for this service.

It should be a rest API that exposes the required endpoints.

The company data can either be autogenerated or configurable or dynamically created. (Up to you)

Nice to have 1: Build a frontend for the rest api

Nice to have 2: Use jvm based technologies

### Data definition

* Employee model
  + Name
  + Surname
  + email
  + address
  + salary
  + company id
* Company model
  + Name

### Delivery

A publicly available git repository with a README.md that explains how to compile, build and start the project.

# Test Cases

## **Employees**

|  |  |  |
| --- | --- | --- |
|  | /api/allEmployees | |
| Trigger | Frontend trigger (home page loaded) | |
| Normal Flow | 1. User navigates to webpage 2. /api/getAllEmployees gets called 3. returns all employees | |
| Expected Result | List of employee objects returned | |
| Actual Result | List of employee objects returned | |
|  | /api/getEmployee | |
| Trigger | User wants to update employee from frontend | |
| Normal Flow | User clicks to update employee  /api/getEmployee gets called based on employee  Corresponding employee info is fetched  All details linked to employee is shown | |
| Expected Result | All details linked to employee is shown | |
| Actual Result | All details linked to employee is shown | |
|  | | /api/addEmployee |
| Trigger | | User adds employee from frontend |
| Normal Flow | | User submits employee info in form  /api/addEmployee called with employee info  Data is validated  Valid employee inserted into database |
| Expected Result | | Valid employee inserted into database |
| Actual Result | | Valid employee inserted into database |
|  | | /api/updateEmployee |
| Trigger | | User updates employee from the frontend |
| 1. Normal Flow | | User submits new employee information on form  /api/updateEmployee called to update the Employee’s information  Data is validated  Employee is updated with valid data |
| Expected Result | | Employee is updated with valid data |
| Actual Result | | Employee is updated with valid data |
|  | | /api/deleteEmployee |
| Trigger | | User confirms the deletion of an employee |
| Normal Flow | | User clicks to delete an employee  User confirms the deletion of the employee  /api/deleteEmployee with the relevant employee data is sent  The given employee is deleted from the database |
| Expected Result | | The given employee is deleted from the database |
| Actual Result | | The given employee is deleted from the database |

## **Company**

|  |  |
| --- | --- |
|  | /api/getAllCompanies |
| Trigger | Frontend trigger (home page loaded) |
| Normal Flow | 1. User navigates to webpage 2. /api/getAllCompanies gets called 3. returns all companies |
| Expected Result | List of company objects returned |
| Actual Result | List of company objects returned |
| **Function Name** | /api/getCompanyAverageSalary |
| Trigger | Frontend trigger (home page loaded) |
| Normal Flow | 1. User navigates to webpage 2. /api/getCompanyAverageSalary gets called 3. returns each individual company’s average salary per employee |
| Expected Result | Returns each individual company’s average salary per employee |
| Actual Result | Returns each individual company’s average salary per employee |