

# REPORT ON REST BASED WEB APPLICATION USING SPRINGBOOT

## Problem Statement:

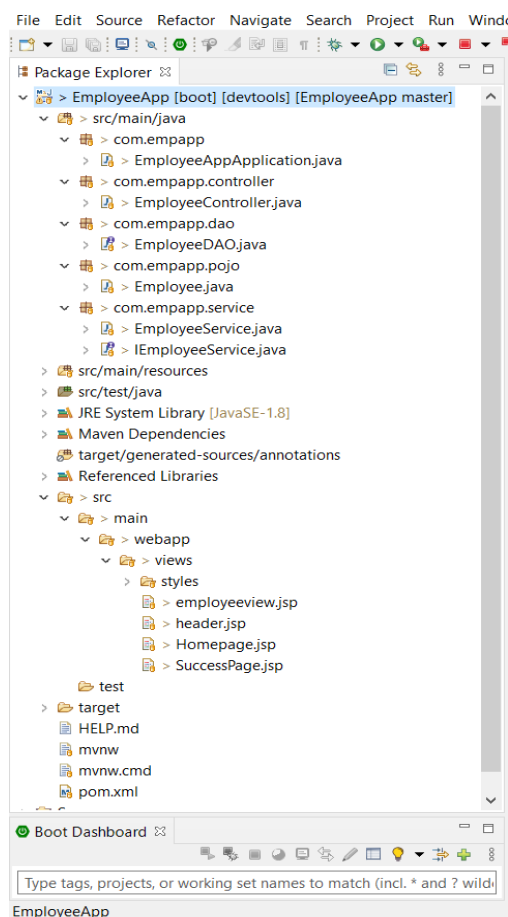
1. Create a form where a user/employee has to register himself/herself.
2. The attributes are: Employee ID, Employee Name, and the Type of the Job of the Employee.
3. When the employee inputs all the attributes, then the details of that particular employee has to be stored in a database.
4. It should display a 'success' message when the employee data has been stored in the database.
5. There should be a button, which on clicking returns the details of all the employees who have been registered.

## Pre-requisites:

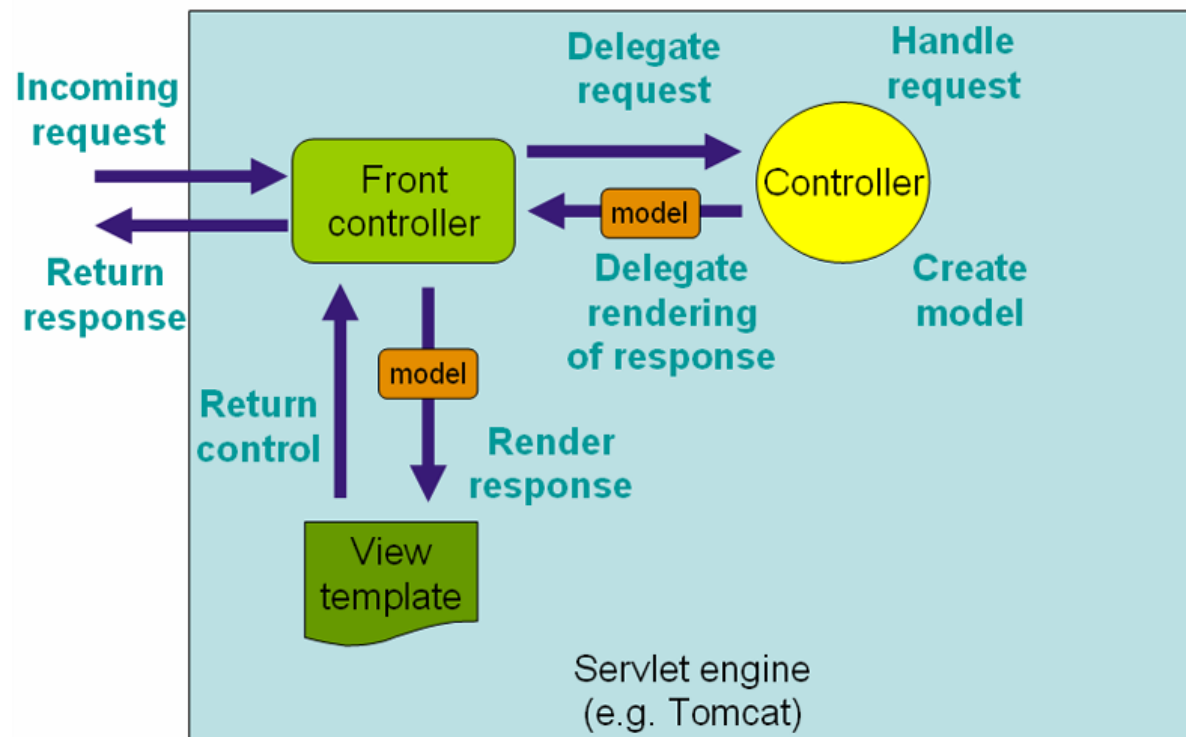
1. Knowledge of Spring framework
2. Knowledge of Java
3. Knowledge of Spring boot Application
4. Database : Any one (Oracle, MySQL, H2, Postgre)

## Solution:

I have developed a Maven Project which consists of the following files:



The following diagram shows the working of the Maven project i.e. the Web Application project that I have created.



According to the above template, we have made our java files, which consists of:

1. One main Java file – SpringBootApplication
2. Controller Java file
3. Database Access/storage file
4. Entity Java file
5. Service Java file and Service Interface file

In the “application.properties” file, we have given our tomcat server port number which will run in the localhost. Also, we have instantiated mvc prefix and suffix so when the controller calls the action, the file of the name provided in controller file creates it with the given suffix as (.jsp). The Oracle database and the Hibernate properties are also included in the mentioned file.

The next file, pom.xml, we are injecting all the dependencies which are more than required for our project. Thanks to SpringBoot which provides a better approach to include all the dependencies at once in a packet. We not need to include everytime. That’s the major advantage of SpringBoot.

In the “/EmployeeApp/src/main/webapp/views”, all the .jsp files have been coded so that when action returns to the controller, it should command the files and show us the corresponding webpages.

## What I have done additionally in this project –

1. I have included form validations.
2. If there is a record present of existing employee id, and new user enters the same id, then it shows “The id is already present, please enter a different id”.
3. Bootstrap has been included for better designing of the web page.
4. “Reset” button is there which resets the values entered by the user.

The UI and the functionality of the REST Based Web Application using SpringBoot has been shown below:

- **Home Page:** It includes the form where the employee has to fill his/her details. The buttons are “Home” which will redirect to the Home Window as shown below. “Reset” button which will reset the details entered by the employee. “Show All Employee Details” which will show all the details of the employees who have been registered successfully.

localhost:9091

Home

Add New Employee Details

Employee Id  
0

Employee Name

Job Type

Add Employee Reset

Show All Employee Details

localhost:9091

- It shows that Employee Name and Job Type can't be empty. This is a Form Validation.

localhost:9091

Home

Add New Employee Details

Employee Id  
321

Employee Name

Job Type

Add Employee Reset

Job Type cannot be blank.

Employee name cannot be blank.

Show All Employee Details

- It shows that Employee Name and Job Type only include Alphabets and Employee ID only include Integers. This is also a form validation.

Home

Add New Employee Details

Employee Id

321

Employee Name

Akash Jain123

Job Type

Data Analyst321

Add Employee

Reset

Only alphabets are allowed in Job Type

Only alphabets are allowed in Employee Name

Show All Employee Details

- After filling all the correct details, when the employee clicks on the “Add Employee” button, the details of that particular employee has been stored in the Oracle Database (that I used here). There is also a “Reset” button, so if you want to reset the Name and Job Type, you can just simply click on the mentioned button.

Home

Add New Employee Details

Employee Id

321

Employee Name

Akash Jain

Job Type

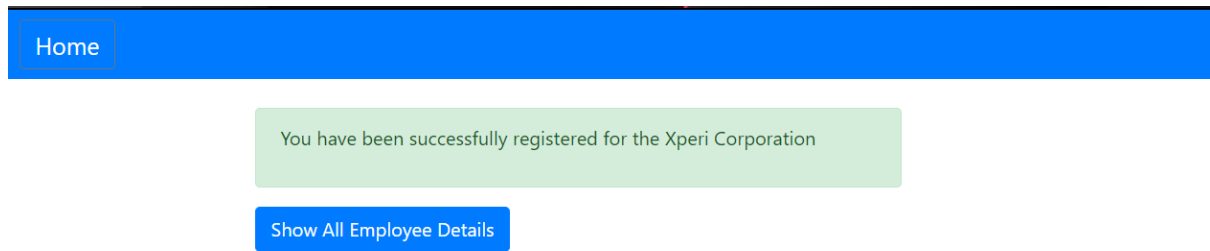
Data Analyst

Add Employee

Reset

Show All Employee Details

- When the employee clicks on the “Add Employee” button, a response message/success message has been displayed that ‘Data Inserted Successfully’.



- By clicking on the “Show All Employee Details”, one can check the details of all the employees registered in the form of tables which has 3 entries: Employee ID, Name and Job Type.

A screenshot of a web application interface. At the top, there is a blue navigation bar with a 'Home' button. Below the navigation bar, there is a table with 3 columns: Employee ID, Name, and Job Type. The table contains 3 rows of data.

Employee ID	Name	Job Type
101	Sushant Singh	Actor
321	Akash Jain	Data Analyst
100	Shubham Jain	Software Dev Intern

localhost:9091

The database working when the mentioned data is inserted:

Given data:

- Employee ID: 321
- Employee Name: Akash Jain
- Job Type: Data Analyst

➤ Before insertion of data, the database looks like this:

Oracle SQL Developer : C:\Users\Shubham Jain\Downloads\Springboot Web Application Project\Test.sql

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists 'Oracle Connections', 'Test', 'Oracle NoSQL Connections', and 'Database Schema Service Connections'. The 'Test' connection is selected. The 'SQL Worksheet' pane shows a query: `select * from employee_data;`. The 'Query Result' pane displays the results of the query, showing 3 rows fetched in 0.003 seconds. The results are as follows:

EMP_ID	EMP_NAME	JOB_TYPE
1	101 Sushant Singh	Actor
2	99 Sonalika Porwal	Cyber Security
3	100 Shubham Jain	Software Dev Intern

The 'Messages - Log' pane at the bottom is empty.

➤ When the data is inserted, the database looks like this and Commit the changes.

Oracle SQL Developer : C:\Users\Shubham Jain\Downloads\Springboot Web Application Project\Test.sql

The screenshot shows the Oracle SQL Developer interface after inserting a new row. The 'SQL Worksheet' pane shows the same query: `select * from employee_data;`. The 'Query Result' pane displays the results of the query, showing 4 rows fetched in 0.003 seconds. The results are as follows:

EMP_ID	EMP_NAME	JOB_TYPE
1	101 Sushant Singh	Actor
2	321 Akash Jain	Data Analyst
3	99 Sonalika Porwal	Cyber Security
4	100 Shubham Jain	Software Dev Intern

The 'Messages - Log' pane at the bottom shows the following message: `Test.sql*: 26 Jul, 2020 11:54:07 PM: Commit successful`

**GitHub Link:** <https://github.com/Shubham11Jain/XperiSpringbootWebApplication>

**Created by:**  
**Shubham Jain**