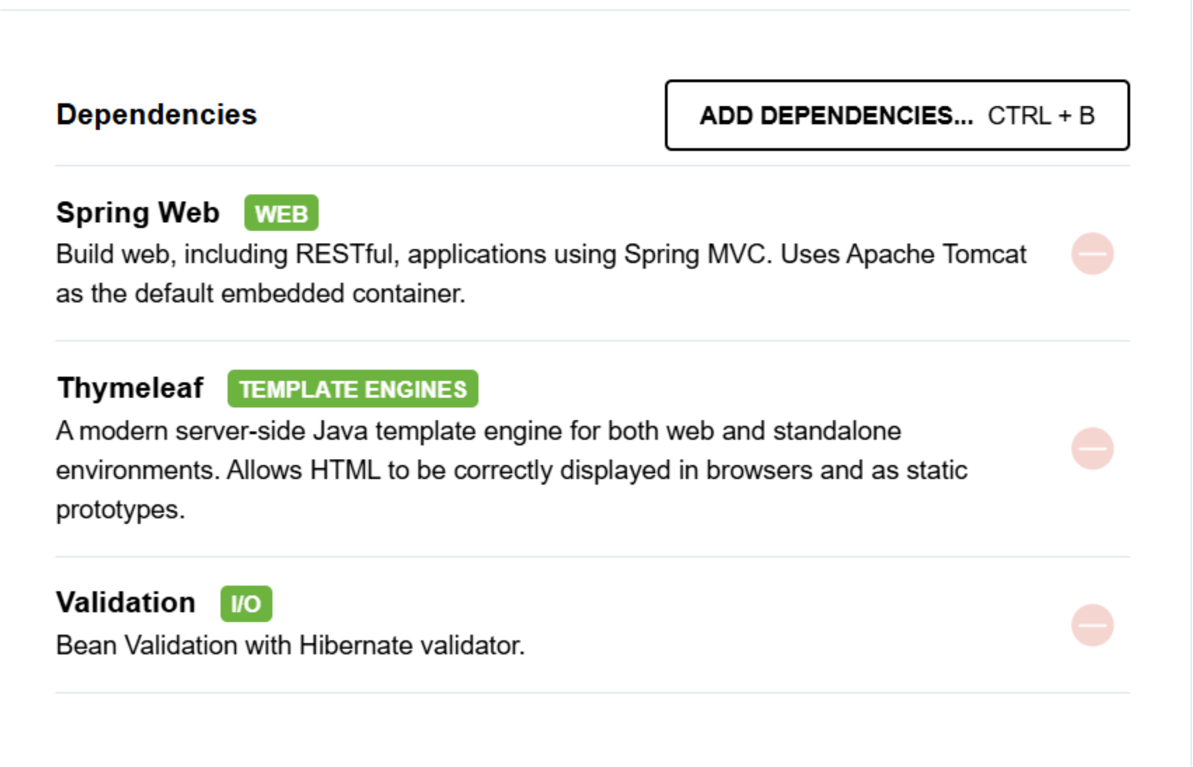
Thymeleaf PostMapping Validation Form

(Bean Validation with Hibernate validator.)

# Use Intellij Idea:

Step :1 Initialize : Spring Initializr :

Injection Dependices Pom.xml. And add the Dependency:



Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Thymeleaf TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

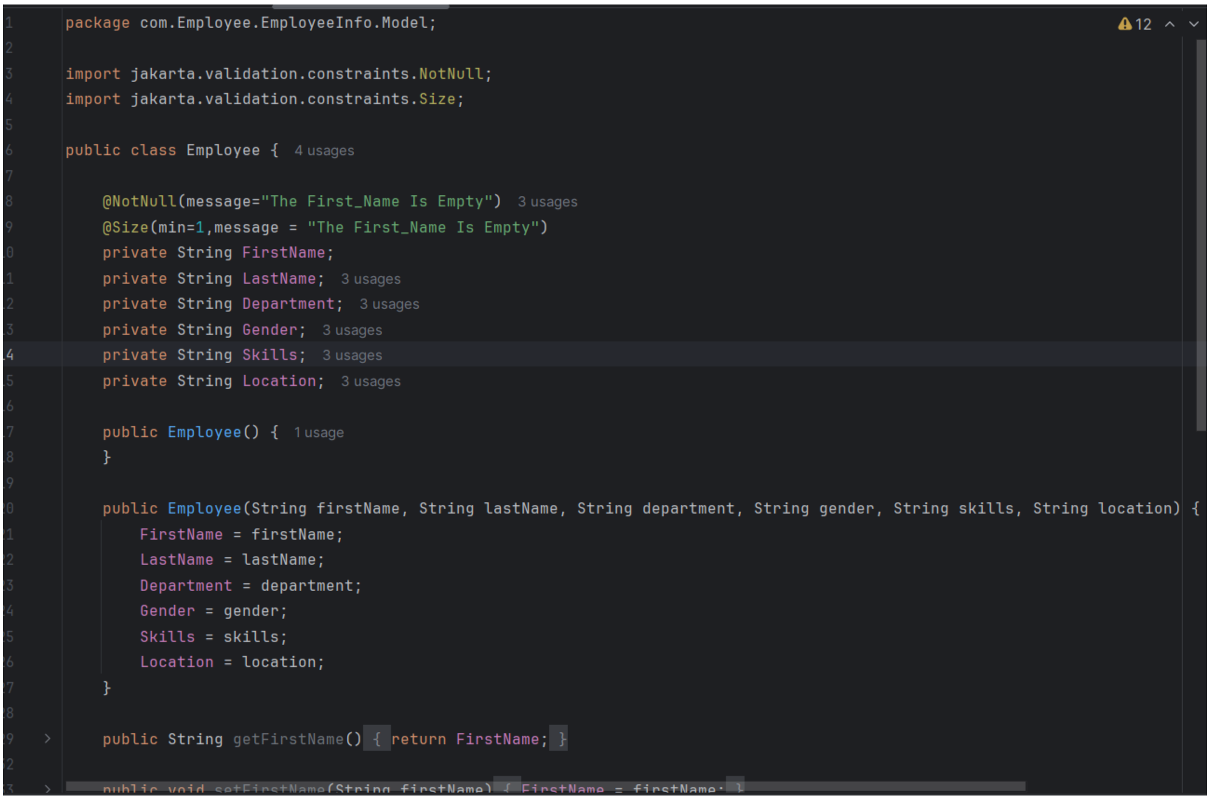
Validation I/O

Bean Validation with Hibernate validator.

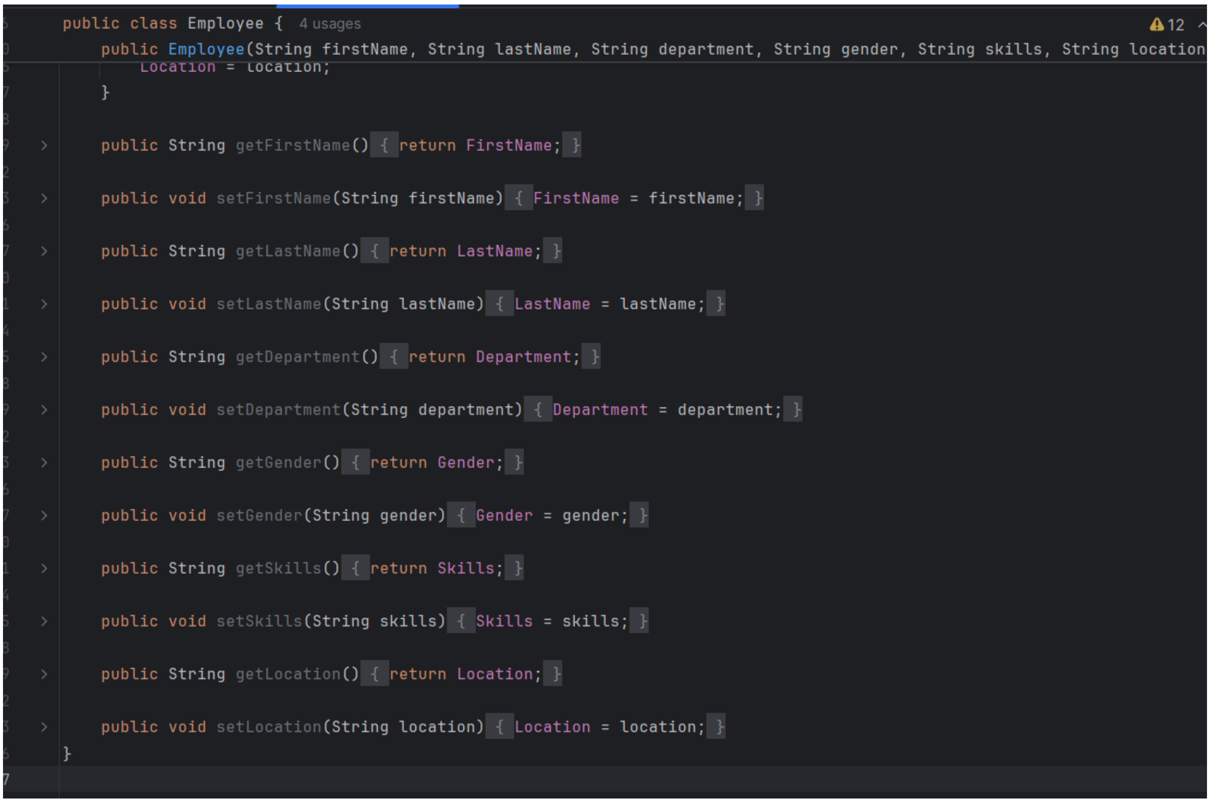
Step 2 : Create a Enitiy Class :

Create java class : use the add employee details details :

Use annotation and Default Constructor and Arguments Constructor and Getter And Setter.



```
package com.Employee.EmployeeInfo.Model;

import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;

public class Employee {  4 usages

    @NotNull(message="The First_Name Is Empty")  3 usages
    @Size(min=1,message = "The First_Name Is Empty")
    private String FirstName;
    private String LastName;  3 usages
    private String Department;  3 usages
    private String Gender;  3 usages
    private String Skills;  3 usages
    private String Location;  3 usages

    public Employee() {  1 usage
    }

    public Employee(String firstName, String lastName, String department, String gender, String skills, String location) {
        FirstName = firstName;
        LastName = lastName;
        Department = department;
        Gender = gender;
        Skills = skills;
        Location = location;
    }

    public String getFirstName() { return FirstName; }

    public void setFirstName(String firstName) { FirstName = firstName; }
```

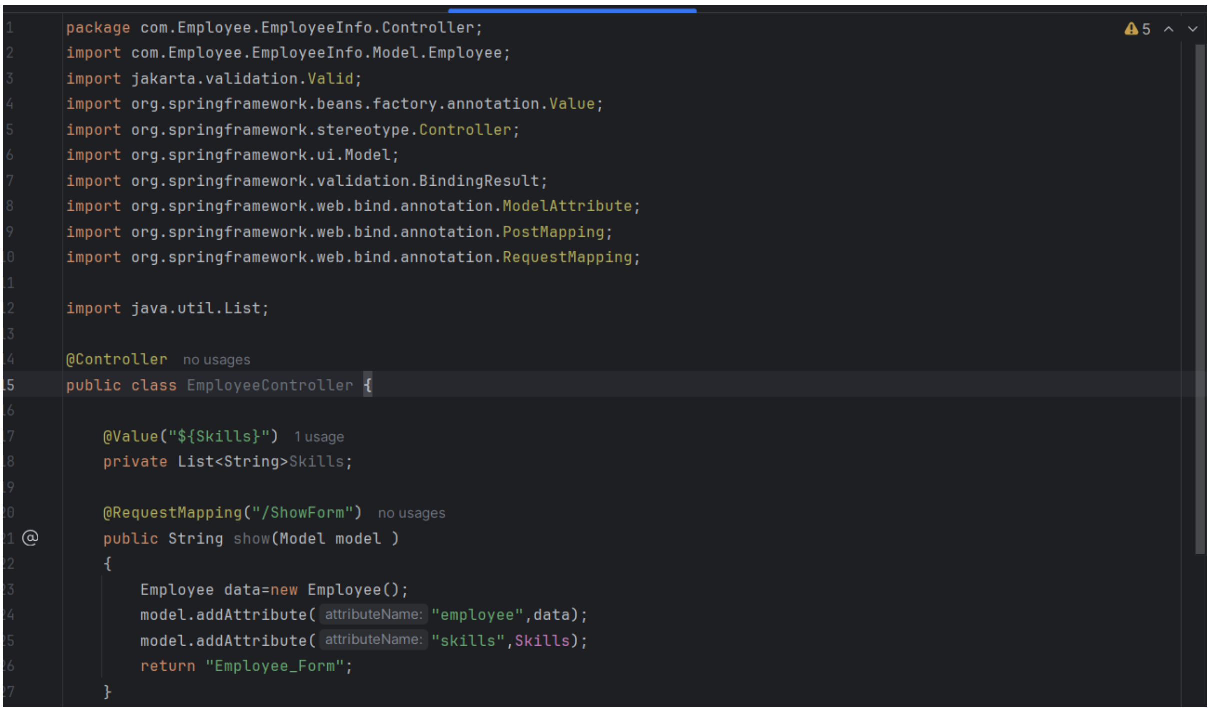


```
public class Employee {  4 usages
    public Employee(String firstName, String lastName, String department, String gender, String skills, String location
        Location = location;
    }

    public String getFirstName() { return FirstName; }

    public void setFirstName(String firstName) { FirstName = firstName; }

    public String getLastName() { return LastName; }

    public void setLastName(String lastName) { LastName = lastName; }

    public String getDepartment() { return Department; }

    public void setDepartment(String department) { Department = department; }

    public String getGender() { return Gender; }

    public void setGender(String gender) { Gender = gender; }

    public String getSkills() { return Skills; }

    public void setSkills(String skills) { Skills = skills; }

    public String getLocation() { return Location; }

    public void setLocation(String location) { Location = location; }
}
```

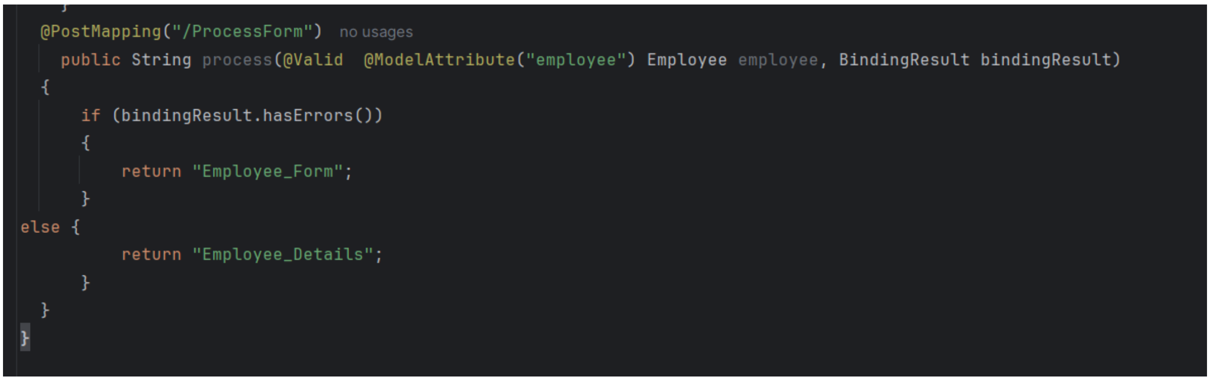
Step :3 Controller Class :

Request Mapping Using show the details and then Object create a class use Data and skills add the details.

“ Post Mapping using to send the details Store the Form”.

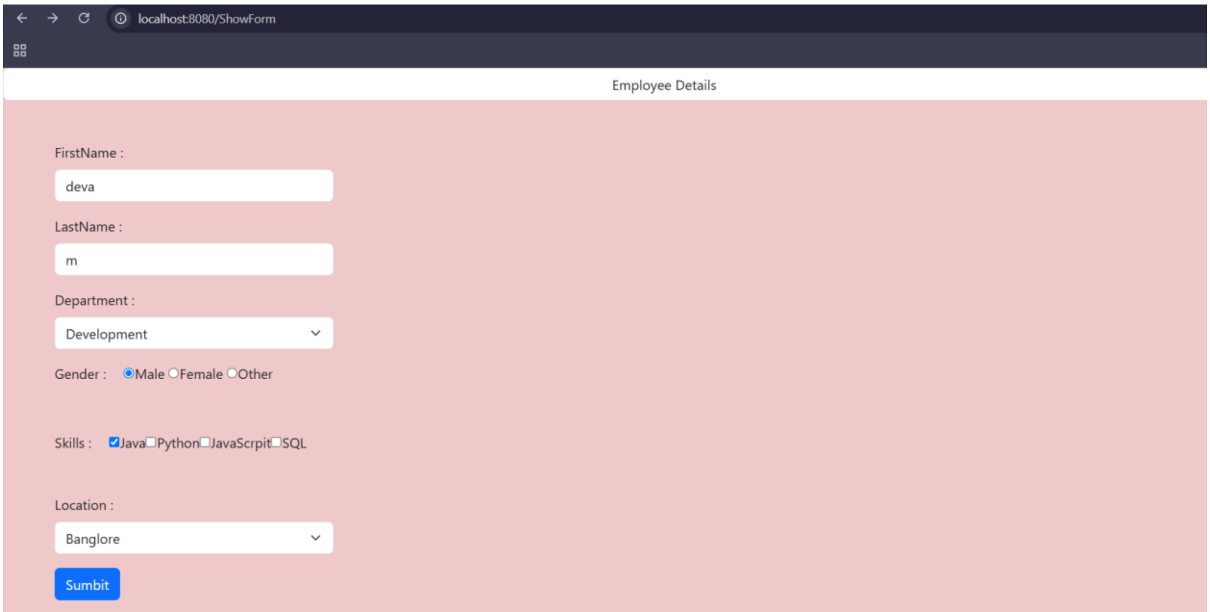


```
package com.Employee.EmployeeInfo.Controller;
import com.Employee.EmployeeInfo.Model.Employee;
import jakarta.validation.Valid;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import java.util.List;

@Controller  no usages
public class EmployeeController {

    @Value("${Skills}")  1 usage
    private List<String>Skills;

    @RequestMapping("/ShowForm")  no usages
    public String show(Model model )
    {
        Employee data=new Employee();
        model.addAttribute( attributeName: "employee",data);
        model.addAttribute( attributeName: "skills",Skills);
        return "Employee_Form";
    }
```



```
   @PostMapping("/ProcessForm")  no usages
     public String process(@Valid  @ModelAttribute("employee") Employee employee, BindingResult bindingResult)
   {
       if (bindingResult.hasErrors())
       {
           return "Employee_Form";
       }
 else {
           return "Employee_Details";
       }
   }
}
```

Output:.



localhost:8080/ShowForm

Employee Details

FirstName :

deva

LastName :

m

Department :

Development

Gender : ● Male ○ Female ○ Other

Skills : ☑ Java □ Python □ JavaScrpit □ SQL

Location :

Banglore

Next The Store the details in the another save the form.

Welcome To Entered Employee Details

FirstName:

deva

LastName:

m

Department:

Development

Gender:

Male

Skills:

Java

Location:

Banglore