# 🌐Spring Boot Web App (MVC + JSP) — Complete Beginner-Friendly Guide

This note explains how to build a **Spring Boot Web Application using Spring MVC + JSP**. You will learn:

✅How Spring MVC works (Controller → View)

✅`@Controller` vs `@RestController`

✅`@RequestMapping`, `@GetMapping`, `@PostMapping`

✅How to send data **from Controller → JSP**

✅How to read **request parameters**

✅How to configure JSP support (because Spring Boot does NOT support JSP by default)

---

# 🚳1. Spring Boot Web vs Spring Boot REST

| Feature | Spring Boot Web (MVC + JSP) | Spring Boot REST |
|---|---|---|
| Controller type | `@Controller` | `@RestController` |
| Response | JSP/HTML pages | JSON response |
| View rendering | Yes | No |
| Use cases | Websites, Forms | APIs, Mobile apps |

**In this guide, we use `@Controller` because we are returning JSP pages.**

---

# 🧳2. Project Setup (pom.xml)

Add these dependencies:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
```

```
<dependency>
    <groupId>jakarta.servlet</groupId>
    <artifactId>jakarta.servlet-api</artifactId>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
</dependency>
```

✔️ `tomcat-embed-jasper` → enables JSP for embedded Tomcat

✔️ `jstl` → enables `<c:forEach>` and JSTL tags

---

# ⤴️3. Folder Structure

```
src/main/java/com.example.demo/
    controllers/
        HomeController.java

src/main/webapp/WEB-INF/views/
    home.jsp
    result.jsp

src/main/resources/application.properties
```

---

# 🛑4. Configure JSP in `application.properties`

```
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

This tells Spring:

```
return "home"  → loads  /WEB-INF/views/home.jsp
```

---

# 🆔 5. Creating Controller

### ☸️ Example: Opening a JSP page

```
@Controller
public class HomeController {

    @GetMapping("/")
    public String home() {
        return "home"; // loads home.jsp
    }
}
```

✔️ `@Controller` → returns JSP name

❌ `@RestController` would return text → NOT JSP

---

# ◀️ 6. Reading Request Parameters

Example: Form → Controller

## 🟡 home.jsp

```
<form action="add" method="get">
    Enter number 1: <input type="text" name="num1"><br>
    Enter number 2: <input type="text" name="num2"><br>
    <button type="submit">Add</button>
</form>
```

## 🆔 Controller

```
@GetMapping("/add")
public String add(HttpServletRequest req, Model model) {
    int a = Integer.parseInt(req.getParameter("num1"));
    int b = Integer.parseInt(req.getParameter("num2"));

    int result = a + b;
    model.addAttribute("result", result);

    return "result"; // result.jsp
}
```

✔️ `HttpServletRequest` works in Spring Boot also

✔️ `Model` is used to pass data to JSP

---

# 🕉️7. result.jsp

```
<h2>Addition Result:</h2>
<p>Result is: ${result}</p>
```

JSP uses **Expression Language (EL)**: `${variableName}`

---

# ♋8. How Spring MVC Internally Works

```
Browser → /add
      ↓
Embedded Tomcat (starts on port 8080)
      ↓
DispatcherServlet (main servlet)
      ↓
HandlerMapping (finds controller)
      ↓
HomeController.add()
      ↓
Return view → "result"
      ↓
InternalResourceViewResolver
      ↓
Loads: /WEB-INF/views/result.jsp
```

---

# 🔁9. Redirect vs Forward

### ✔️Forward

```
return "home";
```

Controller → JSP directly

### ✔️Redirect

```
return "redirect:/hello";
```

Client is redirected and URL changes.

---

## 🔯10. Sending Model Data

**Option 1:** `Model`

```
model.addAttribute("name", "Shiv");
```

**Option 2:** `ModelAndView`

```
ModelAndView mv = new ModelAndView("result");
mv.addObject("marks", 90);
```

**Option 3:** `@RequestParam`

```
public String add(@RequestParam int num1,
                  @RequestParam int num2,
                  Model m) {
    m.addAttribute("result", num1 + num2);
    return "result";
}
```

---

## ☑️11. Full Working Example (Spring Boot Web JSP)

### 🕸️Controller

```
@Controller
public class HomeController {

    @GetMapping("/")
    public String home() {
        return "home";
    }

    @PostMapping("/submit")
    public String submit(@RequestParam String username, Model model) {
        model.addAttribute("name", username);
        return "result";
```

```
    }
}
```

## 🛞home.jsp

```
<form action="submit" method="post">
    Enter name: <input type="text" name="username">
    <button type="submit">Submit</button>
</form>
```

## 🛞result.jsp

```
<h2>Hello, ${name}</h2>
```

---

# ‼️ Summary

**You learned:**

✔️How to configure JSP in Spring Boot ✔️How to use `@Controller` ✔️How Spring MVC resolves JSP pages ✔️How to read request parameters ✔️How to send data from controller → JSP ✔️How `DispatcherServlet` orchestrates everything

---

If you want, I can now create:

🈸**A full Spring Boot Web (JSP) Student Management App**

or

🈲**A note comparing JSP vs Thymeleaf vs React (best view layer)**.