

Spring Boot WAR Project Setup with JSP View (Step-by-Step Guide)

✓ Project Creation using Spring Initializr

1. Open **Spring Initializr** (<https://start.spring.io/>).
 2. **Select the following configurations:**
 - **Project:** Maven
 - **Language:** Java
 - **Spring Boot:** *(Leave default version)*
 - **Packaging:** WAR
 - **Java Version:** Choose 17, 21, or 24 (based on your setup)
 3. **Project Metadata:**
 - **Name:** gtecsiteapp (example)
 - **Artifact:** gtecsiteapp
 - **Group ID:** Use reverse of company URL, e.g., com.gtecsite
 - **Package Name:** com.gtecsite.gtecsiteapp
 4. Click **Next** and in **Dependencies**, add:
 - Spring Web
 - Spring Boot DevTools
 5. Click **Finish** and allow project to load in your IDE (STS or Eclipse).
-

📁 Create Controller Class

1. In `src/main/java`, navigate to package `com.gtecsite.gtecsiteapp`.
2. Right-click the package → **New** → **Class**
3. In the dialog:
 - **Package:** `com.gtecsite.gtecsiteapp.controller`
 - **Name:** `HomeController`
4. Click **Finish**.

Your project will now have the controller class located at:

```
src/main/java/com/gtecsite/gtecsiteapp/controller/HomeController.java
```

Edit HomeController.java

```
package com.gtecsite.gtecsiteapp.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class HomeController {

    @GetMapping({"/", "Index"})
    public String IndexPage() {
        return "Index";
    }

    @GetMapping("/About")
    public String AboutPage() {
        return "About";
    }
}
```

Use Ctrl + Shift + O to auto-import required classes.

Create JSP Pages

1. In src/main/webapp, create the following structure:

```
WEB-INF/
├── views/
│   └── jsp/
│       ├── Index.jsp
│       └── About.jsp
```

2. Sample content for Index.jsp:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Home Page</title>
</head>
<body>
    <h2>Welcome to GTEC Site Application</h2>
</body>
</html>
```

Add Embedded Jasper Dependency

1. Open [Maven Repository](https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper).
2. Search for: tomcat-embed-jasper
3. Choose a version (e.g., 11.0.6) and copy the Maven snippet:

```
<!-- https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper -->
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <version>11.0.6</version>
</dependency>
```

4. Paste it under <dependencies> section in pom.xml.
-

Configure application.properties

Edit src/main/resources/application.properties and add:

```
server.port=8087
spring.mvc.view.prefix=/WEB-INF/views/jsp/
spring.mvc.view.suffix=.jsp
```

▶ Run the Application

1. In **Project Explorer**, right-click the project → **Run As** → **Run Configurations**
 2. Create a new Maven Build configuration:
 - **Name:** (Optional)
 - **Base Directory:** Select the current project
 - **Goals:**
 - clean spring-boot:run -X
 - Click **Apply** → **Run**
-

Expected Console Output

Look for:

```
INFO ... LiveReload server is running on port 35729
INFO ... Tomcat started on port 8087 (http) with context path '/'
```

🌐 Accessing Your Web Pages

- Open browser and go to:

```
http://localhost:8087  
or  
http://localhost:8087/Index
```

- To access About page:

```
http://localhost:8087/About
```

📖 Tips

- You can add more .jsp files like `Contact.jsp`, `Services.jsp`, etc. inside `views/jsp/`.
 - Add corresponding `@GetMapping()` methods in `HomeController.java` for routing.
 - Always save all files before running the project.
-

🔗 **Document Prepared for Copying into Microsoft Word with Web Links, JSP Setup, Maven Dependencies, and Console Verification**