

Step-by-Step: Display Multiple HTML Pages in Spring Boot

🔗 Step 1: Create Project

Go to **Spring Initializr** (<https://start.spring.io/>)

Fill:

- Project: Maven
- Language: Java

As shown below



Project

- Gradle - Groovy
- Gradle - Kotlin
- Maven

Language

- Java
- Kotlin
- Groovy

Spring Boot

- 4.1.0 (SNAPSHOT)
- 4.1.0 (M3)
- 4.0.5 (SNAPSHOT)
- 4.0.4
- 3.5.13 (SNAPSHOT)
- 3.5.12

Project Metadata

Group

Artifact

Package name

Packaging Jar War

Configuration Properties YAML

Java 26 25 21 17

And Add Following Dependencies

- Dependencies:
 - Spring Web
 - Thymeleaf

Download & open in IDE (IntelliJ / Eclipse)

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.

□ Step 2: Project Structure

After opening, your project should look like:

```
src
├── main
│   ├── java/com/example/demo
│   │   └── DemoApplication.java
│   └── resources
│       ├── templates
│       └── application.properties
```



□ Step 3: Create Controller

□ Create a new class: `PageController.java`

`PageController.java` file code:-

```
package com.example.spring_web;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
@Controller
```

```
public class PageController {
```

```
    @GetMapping("/")
```

```
public String home() {  
    return "home";  
}
```

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

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

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

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

```
@GetMapping("/login")
```

```
public String login() {  
    return "login";  
}  
}
```

🔗 Step 4: Create HTML Files

Go to:

```
src/main/resources/templates/
```

Create 6 files:

- home.html
- about.html
- contact.html
- services.html
- blog.html
- login.html

🔗 Step 5: Add Basic HTML (Example)

📄 home.html

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Home</title>  
</head>  
<body>  
    <h1>Home Page</h1>  
  
    <a href="/">Home</a> |  
    <a href="/about">About</a> |  
    <a href="/contact">Contact</a> |  
    <a href="/services">Services</a> |  
    <a href="/blog">Blog</a> |  
    <a href="/login">Login</a>
```

```
</body>
</html>
```

□ about.html

```
<h1>About Page</h1>
<a href="/">Go Home</a>
```

□ contact.html

```
<h1>Contact Page</h1>
<a href="/">Go Home</a>
```

□ services.html

```
<h1>Services Page</h1>
<a href="/">Go Home</a>
```

□ blog.html

```
<h1>Blog Page</h1>
<a href="/">Go Home</a>
```

□ login.html

```
<h1>Login Page</h1>
<a href="/">Go Home</a>
```

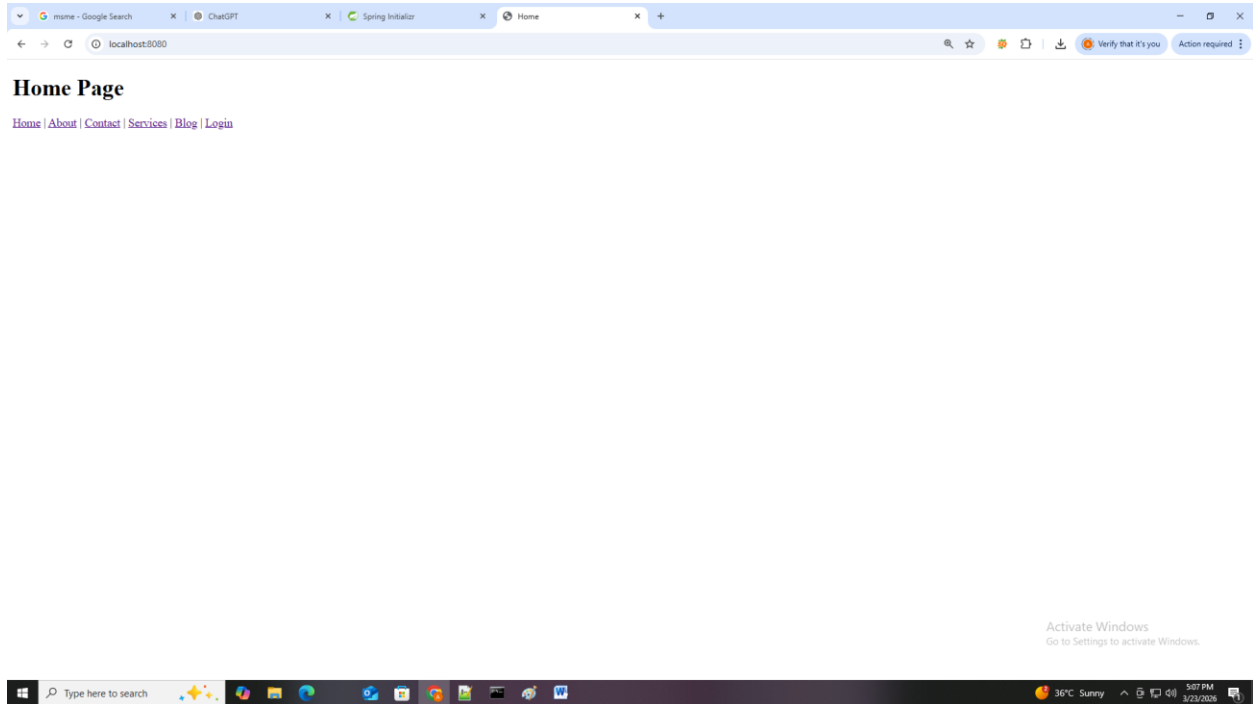
Finally Run it:-

```
D:\java-spring-boot\spring-web>mvn spring-boot:run
```

```
D:\java-spring-boot\spring-web>mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.example:spring-web >-----
[INFO] Building 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
```

Open browser and visit :-

<http://localhost:8080/>



<http://localhost:8080/about>

