

[Home](#) > [member-paid](#) > 03: Spring boot MVC tutorial with thymeleaf & bean validation

# 03: Spring boot MVC tutorial with thymeleaf & bean validation

Posted on July 11, 2016 by Arulkumaran Kumaraswamipillai

**Step 1:** Create a “simple-spring-mvc” maven project.

```
1
2 C:\Users\java-_000\projects>mvn archetype:generat
3
```

**Step 2:** Import the project into eclipse. **File > Import -> Existing Maven Projects** and select “C:\Users\java-\_000\projects\simple-spring-mvc”.

**Step 3:** The **pom.xml** file should look like:

```
1
2 <project xmlns="http://maven.apache.org/POM/4.0.
3     xsi:schemaLocation="http://maven.apache.org/
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>com.mytutorial</groupId>
6     <artifactId>simple-spring-boot</artifactId>
7     <packaging>jar</packaging>
8     <version>1.0-SNAPSHOT</version>
```

## 600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

- ✚ Ice Breaker Interview
- ✚ Core Java Interview C
- ✚ JEE Interview Q&A (3
- ✚ Pressed for time? Jav
- ✚ SQL, XML, UML, JSC
- ✚ Hadoop & BigData Int
- ✚ Java Architecture Inte
- ✚ Scala Interview Q&As
- ✚ Spring, Hibernate, & I
- ✚ Spring (18)
  - ✚ Spring boot (4)
    - ♥ ♦ 11 Spring bc
    - 01: Simple Sp
    - 02: Simple Sp
    - 03: Spring bo
  - ✚ Spring IO (1)
  - ✚ Spring JavaConl
    - 01: ♥ ♦ 13 Spring
    - 01b: ♦ 13 Spring
    - 02: ► Spring DI
    - 03: ♥ ♦ Spring DI
    - 04 ♦ 17 Spring b

```

9      <name>simple-spring-boot</name>
10     <url>http://maven.apache.org</url>
11
12     <parent>
13         <groupId>org.springframework.boot</groupId>
14         <artifactId>spring-boot-starter-parent</artifactId>
15         <version>1.3.6.RELEASE</version>
16     </parent>
17
18     <dependencies>
19         <dependency>
20             <groupId>junit</groupId>
21             <artifactId>junit</artifactId>
22             <scope>test</scope>
23         </dependency>
24
25         <dependency>
26             <groupId>org.springframework.boot</groupId>
27             <artifactId>spring-boot-starter-thymeleaf</artifactId>
28         </dependency>
29
30         <dependency>
31             <groupId>javax.validation</groupId>
32             <artifactId>validation-api</artifactId>
33             <version>1.1.0.Final</version>
34         </dependency>
35     </dependencies>
36
37     <build>
38         <plugins>
39             <plugin>
40                 <groupId>org.springframework.boot</groupId>
41                 <artifactId>spring-boot-maven-plugin</artifactId>
42             </plugin>
43         </plugins>
44     </build>
45
46 </project>
47

```

“spring-boot-starter-thymeleaf” is used instead of “spring-boot-starter-web” as it allows you to use **htmls** instead of JSPs

**Step 4:** POJO (Plain Old Java Object) with annotations for input validation. This is the “Model” of the MVC pattern

```

1
2  package com.mytutorial;
3
4  import javax.validation.constraints.Min;
5  import javax.validation.constraints.NotNull;
6  import javax.validation.constraints.Size;
7
8  public class LicenceForm {
9
10     @Size(min = 2, max = 40)
11     private String name;
12

```

05: ♦ 9 Spring Boot  
06: ♥ Debugging  
07: Debugging S  
Spring loading p  
+ Hibernate (13)  
+ AngularJS (2)  
+ Git & SVN (6)  
+ JMeter (2)  
+ JSF (2)  
+ Maven (3)  
+ Testing & Profiling/Sa  
+ Other Interview Q&A 1  
+ Free Java Interview

## 16 Technical Key Areas

open all | close all

+ Best Practice (6)  
+ Coding (26)  
+ Concurrency (6)  
+ Design Concepts (7)  
+ Design Patterns (11)  
+ Exception Handling (3)  
+ Java Debugging (21)  
+ Judging Experience (1)  
+ Low Latency (7)  
+ Memory Management (1)  
+ Performance (13)  
+ QoS (8)  
+ Scalability (4)  
+ SDLC (6)  
+ Security (13)  
+ Transaction Management (1)

**80+ step by step Java Tutorials**

```

13     @NotNull
14     @Min(16)
15     private Integer age;
16
17     public String getName() {
18         return name;
19     }
20
21     public void setName(String name) {
22         this.name = name;
23     }
24
25     public Integer getAge() {
26         return age;
27     }
28
29     public void setAge(Integer age) {
30         this.age = age;
31     }
32
33     @Override
34     public String toString() {
35         return "LicenceForm [name=" + name + ",
36     }
37 }
38

```

“name” must be between 2 to 40 character long. The “age” must NOT be null and greater than or equal to 16.

**Step 5:** “Controller” of the MVC pattern.

```

1
2 package com.mytutorial;
3
4 import javax.validation.Valid;
5
6 import org.springframework.stereotype.Controller;
7 import org.springframework.validation.BindingResult;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
10
11 @Controller
12 public class LicenceController extends WebMvcConfigurer {
13
14     @RequestMapping(value = "/", method = RequestMethod.GET)
15     public String showForm(LicenceForm licenceForm) {
16         return "licence-form";
17     }
18
19     @RequestMapping(value = "/", method = RequestMethod.POST)
20     public String submitForm(@Valid LicenceForm licenceForm, BindingResult bindingResult) {
21
22         if (bindingResult.hasErrors()) {
23             return "licence-form";
24         }
25
26         return "redirect:/results";
27     }
28 }
29

```

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2\)](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tutorial \(1\)](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Tutorial \(1\)](#)
- [Hibernate Tutorial \(1\)](#)
- [Spring Basics Tutorial \(1\)](#)
- [Spring batch tutorial \(1\)](#)
- [JPA-Hibernate-Spring Tutorial \(1\)](#)
- [Spring Boot Tutorial \(1\)](#)
- [01: Simple Spring Boot Tutorial \(1\)](#)
- [02: Simple Spring Boot Tutorial \(1\)](#)
- [03: Spring boot MVC tutorial with thymeleaf & bean validation \(1\)](#)
- [Spring IO Tutorial \(1\)](#)
- [Spring ORM Tutorial \(1\)](#)
- [Spring other tutorials \(1\)](#)
- [Spring OXM Tutorial \(1\)](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

## 100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [Complete the given code \(1\)](#)
- [Converting from A to B \(1\)](#)
- [Designing your class \(1\)](#)
- [Java Data Structures \(1\)](#)
- [Passing the unit tests \(1\)](#)
- [What is wrong with this code? \(1\)](#)
- [Writing Code Home Assignment \(1\)](#)
- [Written Test Core Java \(1\)](#)
- [Written Test JEE \(1\)](#)

```

30
31     @Override
32     public void addViewControllers(ViewController
33         registry.addViewController("/results").s
34     }
35
36 }
37

```

## How good are your .....?

[open all](#) | [close all](#)

[Career Making Know-](#)  
[Job Hunting & Resum](#)

### Step 6: “Views” of the MVC pattern.

Firstly, the “Get” page stored under  
**“src/main/resources/templates”** folder. You can create the  
 src/main/resources folder by right-clicking “Build Path”, and  
 then **“New Source Folder”**, and then “src/main/resources”.  
 Then add new folder **“templates”** inside.

create **“licence-form.html”** under  
 src/main/resources/templates.

```

1
2 <html>
3
4 <body>
5
6     <form action="#" th:action="@{/}" th:object=
7
8         <table>
9             <tr>
10                 <td>Licensee Name:</td>
11                 <td><input type="text" th:field=
12                 <td th:if="${#fields.hasErrors('
13                     Error</td>
14             </tr>
15
16             <tr>
17                 <td>Licensee Age:</td>
18                 <td><input type="text" th:field=
19                 <td th:if="${#fields.hasErrors('
20                     Error</td>
21             </tr>
22
23             <tr>
24                 <td><button type="submit">Submit
25             </tr>
26         </table>
27     </form>
28 </body>
29
30 </html>
31

```

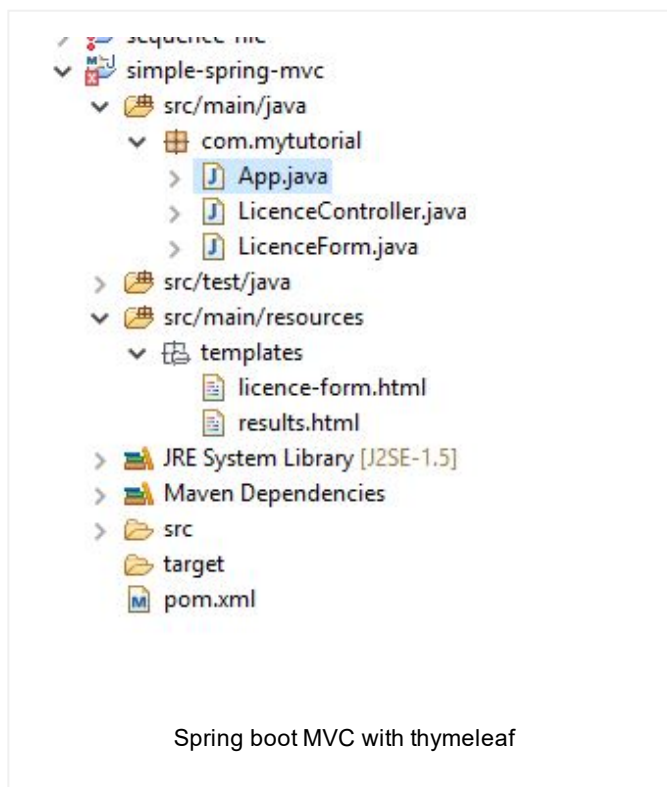
create **“results.html”** under src/main/resources/templates.

```
1
2 <html>
3   <body>You can apply for a license</body>
4 </html>
5
```

**Step 7:** Finally, modify the “**App.java**” as shown below, which can be run within the embedded **Tomcat** server.

```
1
2 package com.mytutorial;
3
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6
7 @SpringBootApplication
8 class App {
9     public static void main(String[] args) {
10         SpringApplication.run(App.class, args);
11     }
12 }
13
14
```

You should now have all the required artifacts.

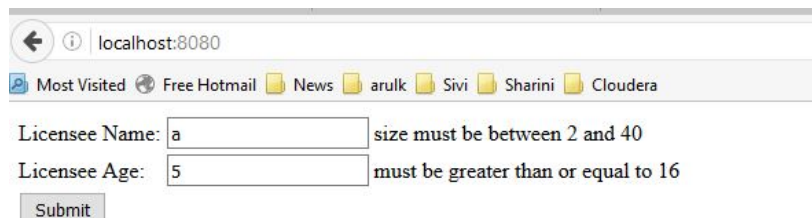


## Run the Spring MVC app

```
1  
2 C:\Users\java-_000\projects\simple-spring-mvc>mvn  
3
```

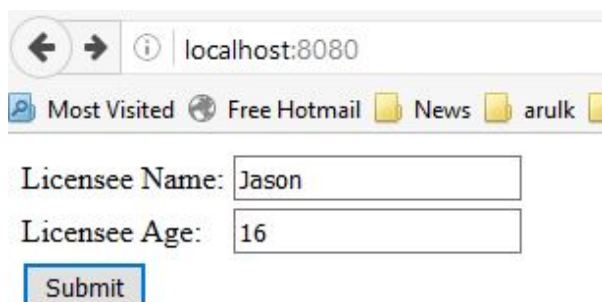
This builds the jar file and deploys to Tomcat on port **8080** on your **localhost**.

## Open in a browser like Firefox



A screenshot of a web browser window showing a form with two input fields. The first field is labeled 'Licensee Name:' and contains the value 'a'. To its right, a validation error message reads 'size must be between 2 and 40'. The second field is labeled 'Licensee Age:' and contains the value '5'. To its right, a validation error message reads 'must be greater than or equal to 16'. Below the fields is a 'Submit' button.

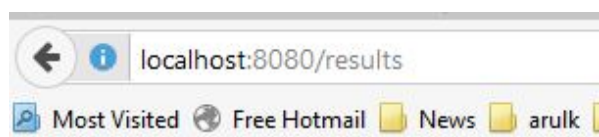
Spring MVC with boot & thymeleaf. Validation errors shown



A screenshot of a web browser window showing the same form as before, but with valid data. The 'Licensee Name' field now contains 'Jason' and the 'Licensee Age' field contains '16'. The 'Submit' button is highlighted with a blue border.

Spring MVC Tutorial – Validation Pass

Click on “**Submit**”



A screenshot of a web browser window showing the results page. The address bar displays 'localhost:8080/results'. The page content includes a message: 'You can apply for a license'.

Spring boot MVC results page.

## Popular Member Posts

## ♦ 11 Spring boot interview questions & answers

904 views

## ♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

816 views

## 001A: ♦ 7+ Java integration styles & patterns interview questions & answers

427 views

## 18 Java scenarios based interview Questions and Answers

407 views

## ♦ 7 Java debugging interview questions & answers

322 views

## 01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

310 views

## 01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

302 views

## ♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

301 views

## ♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

251 views

## 001B: ♦ Java architecture & design concepts interview questions & answers

209 views

0

Like

Share

Tweet

↑

submit

↓

reddit

0

G+1

Share

Bio

Latest Posts



### Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and



often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)



#### About [Arulkumaran Kumaraswamipillai](#)

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via [Amazon.com](https://www.amazon.com) in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site. **945+** paid members. [join my LinkedIn Group](#). [Reviews](#)

◀ 4. ♦ Setting up HBase with zookeeper to be used in Java via maven project

01: Websphere MQ Client API Tutorial – Message Sender ▶

**Posted in** member-paid, Spring boot, Spring Boot Tutorial

## Empowers you to open more doors, and fast-track

### Technical Know Hows

☀ [Java generics in no time](#) ☀ [Top 6 tips to transforming your thinking from OOP to FP](#) ☀ [How does a HashMap internally work? What is a hashing function?](#)  
 ☀ [10+ Java String class interview Q&As](#) ☀ [Java auto un/boxing benefits & caveats](#) ☀ [Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect](#)

### Non-Technical Know Hows

☀ [6 Aspects that can motivate you to fast-track your career & go places](#) ☀ [Are you reinventing yourself as a Java developer?](#) ☀ [8 tips to safeguard your Java career against offshoring](#) ☀ [My top 5 career mistakes](#)



## Prepare to succeed

☀ [Turn readers of your Java CV go from “Blah blah” to “Wow”?](#) ☀ [How to prepare for Java job interviews?](#) ☀ [16 Technical Key Areas](#) ☀ [How to choose from multiple Java job offers?](#)

Select Category ▼

## © Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.