A close up of a logo

Description automatically generated

# Introduction to Spring MVC Lab

FIT5042 Week 7

In this lab we will explore basic spring MVC framework.  
And get prerequisites ready for advanced Spring MVC in order to achieve transaction management.

There is no need of starter files for this lab exercises.

Table of Contents

[Introduction to Spring MVC Lab 1](#_Toc38195417)

[Part 1 : Building a simple Spring MVC application in Intellij 3](#_Toc38195418)

[Create Spring Based New Project 3](#_Toc38195419)

[Web Configuration – web.xml 5](#_Toc38195420)

[Configure servlet config – rename dispatcher-servlet.xml to HelloWeb-servlet.xml 6](#_Toc38195421)

[Configure application context – applicationContext.xml 7](#_Toc38195422)

[Create welcome file directly under web resource - index.jsp 8](#_Toc38195423)

[Create a folder ‘jsp’ under WEB-INF and Create hello.jsp under jsp 8](#_Toc38195424)

[Create ‘java’ Directory under src 9](#_Toc38195425)

[Create controller class com.controller.HelloController under java 10](#_Toc38195426)

[Edit Run Configurations 11](#_Toc38195427)

[Run Output 13](#_Toc38195428)

[Artifacts to submit 14](#_Toc38195429)

[Part 2: Prerequisites for Advance Spring MVC 15](#_Toc38195430)

[Overall Required libraries and dependencies 15](#_Toc38195431)

[1.Maven should be installed in the system. 16](#_Toc38195432)

[2.Application Server – Tomcat should be installed in the system 16](#_Toc38195433)

[3.Database Server – MySQL should be installed 16](#_Toc38195434)

## Part 1 : Building a simple Spring MVC application in Intellij

The basic and quick steps are to understand the web framework only.

Either import the Project file and understand the configs Or follow the steps to build one.

### Create Spring Based New Project

Select spring MVC and Application Server Tomcat (should be pre configured)

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Then it will download some related libraries.

Default project structure after this

A screenshot of a cell phone

Description automatically generated

### Web Configuration – web.xml

Configure the welcome file and servlet engine and set up on load

*<?***xml version="1.0" encoding="UTF-8"***?>*<**web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"  
 version="4.0"**>  
 <**context-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>/WEB-INF/applicationContext.xml</**param-value**>  
 </**context-param**>  
 <**listener**>  
 <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**>  
 </**listener**>  
 <**servlet**>  
 <**servlet-name**>HelloWeb</**servlet-name**>  
 <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  
 <**load-on-startup**>2</**load-on-startup**>  
 </**servlet**>  
 <**servlet-mapping**>  
 <**servlet-name**>HelloWeb</**servlet-name**>  
 <**url-pattern**>/</**url-pattern**>  
 </**servlet-mapping**>  
 <**session-config**>  
 <**session-timeout**>  
 30  
 </**session-timeout**>  
 </**session-config**>  
 <**welcome-file-list**>  
 <**welcome-file**>index.jsp</**welcome-file**>  
 </**welcome-file-list**>   
</**web-app**>

### Configure servlet config – rename dispatcher-servlet.xml to HelloWeb-servlet.xml

Rename dispatcher-servlet.xml to HelloWeb-servlet.xml by right-clicking on dispatcher-servlet.xml, then select Refactor/Rename.

*<?***xml version="1.0" encoding="UTF-8"***?>*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:p="http://www.springframework.org/schema/p"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xmlns:tx="http://www.springframework.org/schema/tx"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd  
 http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd"**>  
  
<**context:component-scan base-package="com.controller"**/>  
<**bean class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>  
 <**property name="prefix" value="/WEB-INF/jsp/"** />  
 <**property name="suffix" value=".jsp"** />  
</**bean**>  
</**beans**>

### Configure application context – applicationContext.xml

*<?***xml version="1.0" encoding="UTF-8"***?>*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:p="http://www.springframework.org/schema/p"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xmlns:tx="http://www.springframework.org/schema/tx"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd  
 http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd"**>  
  
*<!--bean id="propertyConfigurer"  
 class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"  
 p:location="/WEB-INF/jdbc.properties" />  
  
<bean id="dataSource"  
class="org.springframework.jdbc.datasource.DriverManagerDataSource"  
p:driverClassName="${jdbc.driverClassName}"  
p:url="${jdbc.url}"  
p:username="${jdbc.username}"  
p:password="${jdbc.password}" /-->  
  
<!-- ADD PERSISTENCE SUPPORT HERE (jpa, hibernate, etc) -->*</**beans**>

### Create welcome file directly under web resource - index.jsp

<%@ **page contentType**="**text/html;charset=UTF-8**"%>  
<!DOCTYPE **HTML** PUBLIC **"-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"**>  
<**html**>  
<**head**>  
 <**meta http-equiv="Content-Type" content="text/html; charset=UTF-8"**>  
 <**title**>Welcome to Spring Web MVC project</**title**>  
</**head**>  
<**body**>  
 <**p**>Hello! This is the default welcome page for a Spring Web MVC project.</**p**>  
 <**a href="${**pageContext.request.contextPath**}/hello"**>Click Here for HelloController</**a**>  
</**body**>  
</**html**>

### Create a folder ‘jsp’ under WEB-INF and Create hello.jsp under jsp

<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>Hello World</**title**>  
</**head**>  
<**body**>  
<**h2**>**${**message**}**</**h2**>  
</**body**>  
</**html**>

### Create ‘java’ directory (Package) under src

After creating java directory (Package) under src, open Modules settings and select java as the source directory.

A screenshot of a social media post

Description automatically generated

### Create controller class com.controller.HelloController under java

**package** com.controller;  
  
**import** org.springframework.stereotype.Controller;  
**import** org.springframework.web.bind.annotation.RequestMapping;  
**import** org.springframework.web.bind.annotation.RequestMethod;  
**import** org.springframework.ui.ModelMap;  
*/\*\*  
 \*  
 \** ***@author*** *sunil  
 \*/*@Controller  
@RequestMapping(**"/hello"**)  
**public class** HelloController{  
  
 @RequestMapping(method = RequestMethod.***GET***)  
 **public** String printHello(ModelMap model) {  
 model.addAttribute(**"message"**, **"Hello Spring MVC Framework from HelloController!"**);  
 **return "hello"**;  
 }  
}

### Edit Run Configurations

Create a new Tomcat server by clicking the + sign, as shown below.

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

### The above Application context name (e.g. /SpringMVc\_intellij) should match the URL content shown in the above “Edit Run Configurations” section.

### Go to Project Structure, select Artifacts, as shown below.

A screenshot of a social media post

Description automatically generated

### Run Output

Click on the project and Run. You should see index.jsp dispayed

A screenshot of a cell phone

Description automatically generated

and when you click on the HelloController path ‘/hello’

A screenshot of a cell phone

Description automatically generated

### Artifacts to submit

1. Github link to code neatly organized under week 7 folder.

<https://github.com/suniluniversal/springmvc-intro/tree/master/SpringMVC_intellij>

1. Output run screenshots detailing all navigation actions.

## Part 2: Prerequisites for Advanced Spring MVC

### Overall Required libraries and dependencies

Web Client 🡨 🡪 Webserver (configured with spring) 🡨 🡪 Database

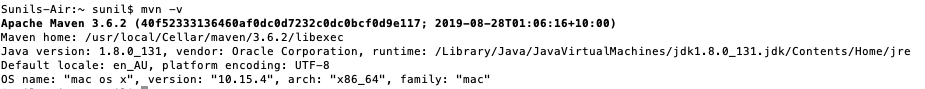
The choice of technologies used are mentioned as below:

|  |  |  |
| --- | --- | --- |
| Java | 8 | Code Language |
| Apache Tomcat | 9.x | Application Server |
| MySQL Server | 8.x | Database Server |
| Intellij | 2018.x | GUI for development |
| Spring framework | 5.x | Web Framework |
| Hibernate framework | 5.x | Data Persistence Implementation Framework |
| Spring Data JPA | 2.1.5 | Java Persistence API |
| Servlet | 3.1 | Core Container to Create Dynamic Web Page |
| Jsp | 2.3.1 | Server side Dynamic Web Page Language |
| Jstl | 1.2 | JSP tag library |
| Maven | 3.x | Build tool to create Executable |

### 1.Maven should be installed in the system.

How to install - <https://maven.apache.org/install.html>

Check if its installed run command **mvn -v**



### 2.Application Server – Tomcat should be installed in the system

How to install and verify installation please follow the link below

<https://www.dev2qa.com/how-to-install-tomcat-in-macos/>

### 3.Database Server – MySQL should be installed

Simply download the top .dmg file for mac os and install - <https://dev.mysql.com/downloads/mysql/>

A screenshot of a social media post

Description automatically generated

Workbench is not compulsory but easy to explore data and schema.

https://dev.mysql.com/downloads/workbench

Point to remember is the root user and the password used.

You can run the command **mysql -u root -p** and prompt password to login into mysql DB