

## Web MVC Framework / Spring MVC Set-Up

The Spring Web model-view-controller (MVC) framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for uploading files.

The default handler is based on the @Controller and @RequestMapping annotations, offering a wide range of flexible handling methods.

With the introduction of Spring 3.0, the @Controller mechanism also allows you to create RESTful Web sites and applications, through the @PathVariable annotation and other features.

### Example:

Create a Maven project by selecting archetype: maven-archetype-webapp.

Write the group id: com.example1

And Artifact id: firstSpringMVCAApp

Set the build path of the project.

Add the following lines of code in **POM.xml**:

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
</properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>5.3.22</version>
        </dependency>
    </dependencies>
```

Go to the path: src→main→webapp→WEB-INF and check web.xml file exist or not.

Create one more xml file for dispatcher servlet with the name of **appSpring-servlet.xml**.

**web.xml** and **appSpring-servlet.xml** files must be in WEB-INF folder.

Add the lines of code in **appSpring-servlet.xml**:

```
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
```

## Web MVC Framework / Spring MVC Set-Up

```
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
    <!-- Adding Support for Component Scan -->
    <context:component-scan
        base-package=" firstSpringMVCAApp " />
    <!-- Configure View Resolver -->
    <bean
        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
            <property name="prefix" value="/WEB-INF/jspPages/" />
            <property name="suffix" value=".jsp" />
        </bean>
    </beans>
```

Create a folder within WEB-INF with the name of **jspPages**. Inside this folder (jspPages) create a JSP page with the name of **a1.jsp** and add a line of code in body tag: **<h1>Hello A1 JSP</h1>**

Add the following lines of code in web.xml:

```
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
    <display-name>Archetype Created Web Application</display-name>
    <servlet>
        <servlet-name>SpringSample</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/ appSpring-servlet.xml </param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>SpringSample</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
</web-app>
```

Right Click on src/main/java and create one package with the name of **firstSpringMVCAApp**. Create one java class within **firstSpringMVCAApp** package with the name of **HomeController.java** and add the lines of code:

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
@Controller
```

## Web MVC Framework / Spring MVC Set-Up

```
public class HomeController {  
    @RequestMapping(value = "/", method = RequestMethod.GET)  
    public String home() {  
        return "a1";  
    }  
}
```