

## Spring MVC

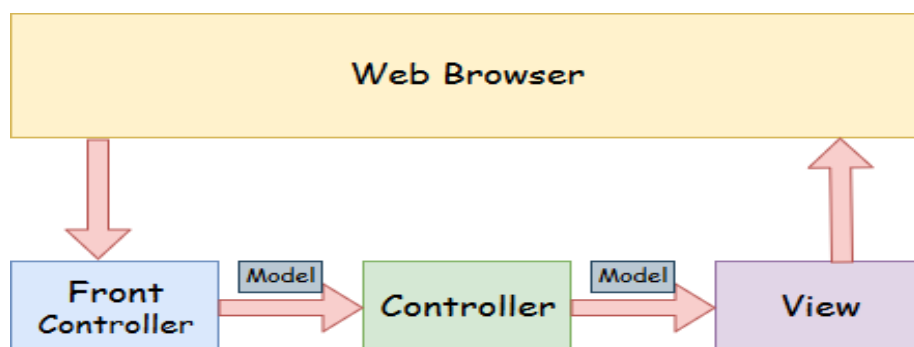
---

- A Spring MVC is a Java framework which is used to build web applications.
- It follows the Model-View-Controller design pattern.
- It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

## Spring MVC

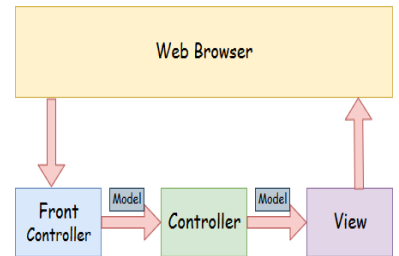
---

- A Spring MVC provides an elegant solution to use MVC in spring framework by the help of **DispatcherServlet**.
- Here, **DispatcherServlet** is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.

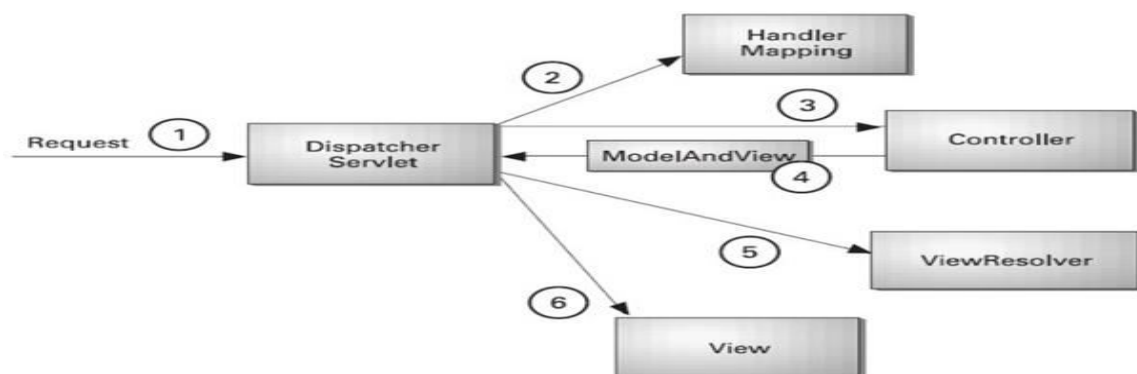


# Spring MVC

- **Model** - A model contains the data of the application. A data can be a single object or a collection of objects.
- **Controller** - A controller contains the business logic of an application. Here, the `@Controller` annotation is used to mark the class as the controller.
- **View** - A view represents the provided information in a particular format. Generally, JSP+JSTL is used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thymeleaf and FreeMarker.
- **Front Controller** - In Spring Web MVC, the `DispatcherServlet` class works as the front controller. It is responsible to manage the flow of the Spring MVC application.



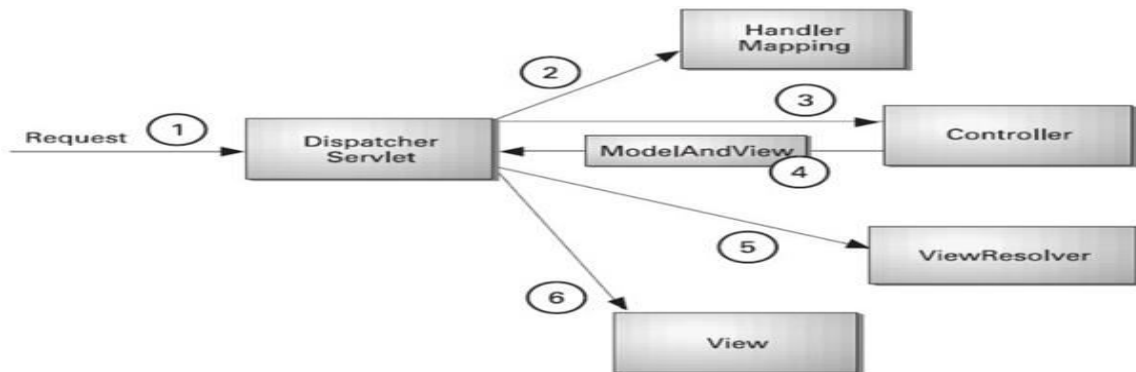
## Flow of Spring Web MVC



- All the incoming request is intercepted by the `DispatcherServlet` that works as the front controller.
- The `DispatcherServlet` gets an entry of handler mapping from the XML file and forwards the request to the controller.

# Flow of Spring Web MVC

---



- The controller returns an object of ModelAndView.
- The DispatcherServlet checks the entry of view resolver in the XML file and invokes the specified view component.

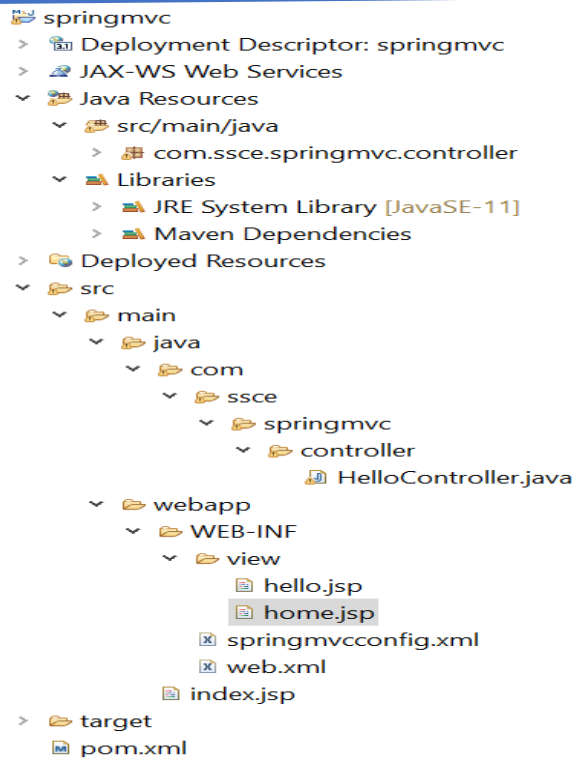
## Creating Spring Web MVC project – XML based

---

- Create Maven project with **org.apache.maven.archetypes – maven-archetype-webapp**
- Add dependencies
- Provide the entry of controller in the web.xml file
- Define the bean in the separate XML file
- Create the controller class
- Display the message in the JSP page
- Start the server and deploy the project

# Creating Spring Web MVC project – XML based

## Directory Structure of Spring MVC



# Creating Spring Web MVC project – XML based

- Add dependencies

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.12</version>
</dependency>
<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/javax.servlet.jsp/javax.servlet.jsp-api -->
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.3</version>
    <scope>provided</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/javax.servlet.jsp.jstl/javax.servlet.jsp.jstl-api -->
<dependency>
    <groupId>javax.servlet.jsp.jstl</groupId>
    <artifactId>javax.servlet.jsp.jstl-api</artifactId>
    <version>1.2.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.taglibs/taglibs-standard-impl -->
<dependency>
    <groupId>org.apache.taglibs</groupId>
    <artifactId>taglibs-standard-impl</artifactId>
    <version>1.2.5</version>
    <scope>runtime</scope>
</dependency>
```

# Creating Spring Web MVC project – XML based

---

- Provide the entry of controller in the web.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-
app_3_1.xsd" id="WebApp_ID" version="3.1">

    <display-name>Spring MVC demo</display-name>

    <absolute-ordering />

    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>index.html</welcome-file>
    </welcome-file-list>

    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>
            org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>
                /WEB-INF/springmvcconfig.xml
            </param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

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

# Creating Spring Web MVC project – XML based

---

- Define the bean in the separate XML file

```
<?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:context="http://www.springframework.org/schema/context"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd">

    <!-- Add support for component scanning -->
    <context:component-scan base-package="com.ssce.springmvc.controller" />

    <!-- Add support for conversion, formatting and validation support -->
    <mvc:annotation-driven/>

    <!-- Define Spring MVC view resolver -->
    <bean
        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/view/" />
        <property name="suffix" value=".jsp" />
    </bean>
</beans>
```

# Creating Spring Web MVC project – XML based

---

- Create the controller class

```
package com.ssce.springmvc.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HelloController {

    @GetMapping("/hello")
    public String getHello(Model model) {
        model.addAttribute("message", "Rajesh Shet");
        return "hello";
    }

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

# Creating Spring Web MVC project – XML based

---

- Create view ie JSP page (hello.jsp)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Hello world</h1>
<h3>Message from controller ${message}</h3>
</body>
</html>
```

# Creating Spring Web MVC project – XML based

---

- Create view ie JSP page (home.jsp)

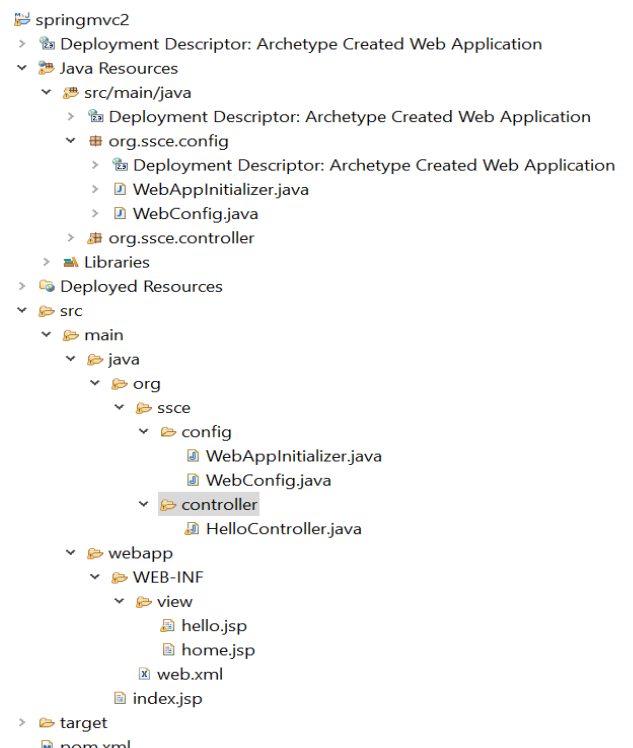
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Home component</h1>
</body>
</html>
```

# Creating Spring Web MVC project – Java based

---

## • Project structure

- **Web.XML is replaced by WebAppInitializer.java**
- **SpringConfig.xml is replaced by WebConfig.java**



# Creating Spring Web MVC project – Java based

---

## WebAppInitializer.java

```
package org.ssce.config;

import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

public class WebAppInitializer extends AbstractAnnotationConfigDispatcherServletInitializer{

    @Override
    protected Class<?>[] getRootConfigClasses() {

        return new Class[] {WebConfig.class};
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {

        return null;
    }

    @Override
    protected String[] getServletMappings() {

        return new String[] {"/"};
    }

}
```

# Creating Spring Web MVC project – Java based

---

## WebConfig.java

```
@Configuration
@ComponentScan(basePackages="org.ssce")
@EnableWebMvc
public class WebConfig {

    @Bean
    public ViewResolver getViewResolver() {
        return new InternalResourceViewResolver("/WEB-INF/view/", ".jsp");
    }

}
```