Spring MVC-

In MVC, the whole functionality is divided into three parts as-model, view

and controller

Model- It can be an object or collection of objects which basically contains the data of the application which consist of POJO.

View- It is used for displaying the information to the user.

Controller- provides information between model and view. It contains the logical part of the application. *@Controller* annotation is used to mark that class as controller.

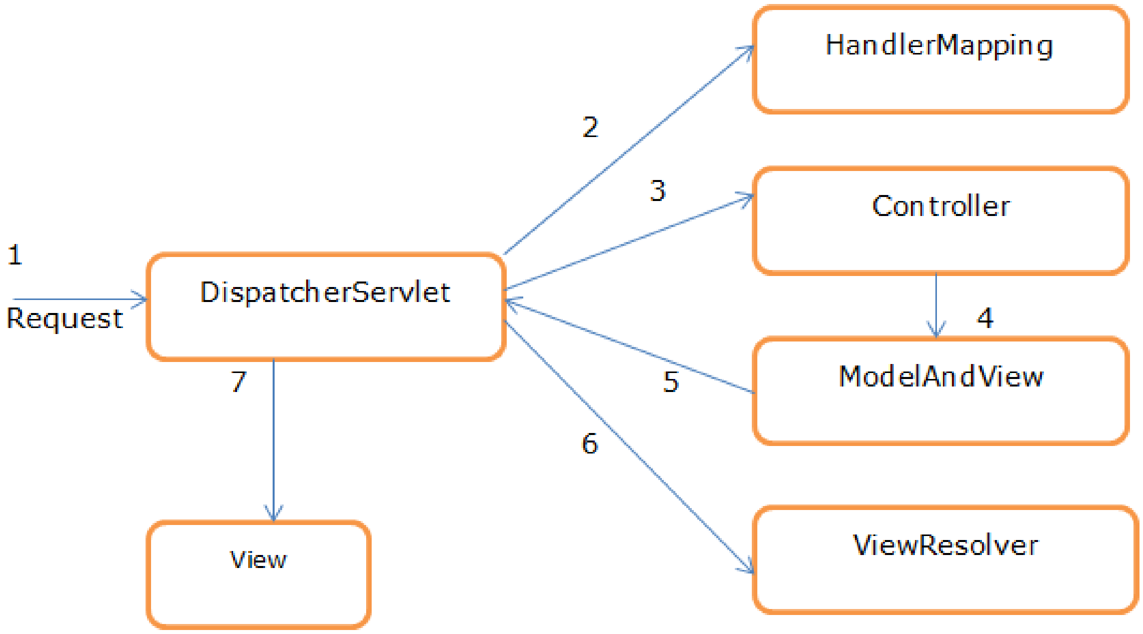


Fig- Spring MVC Flow

Description-

Step 1:

When browser sent the request will be received by DispatcherServlet.

Step 2:

DispatcherServlet will take the help of HandlerMapping and get to know the Controller class name associated with the given request.

Step 3:

So request transfer to the Controller, and then controller will process the request by executing appropriate methods and returns ModeAndView object (contains *Model* data and *View* name) back to the DispatcherServlet.

Step 4:

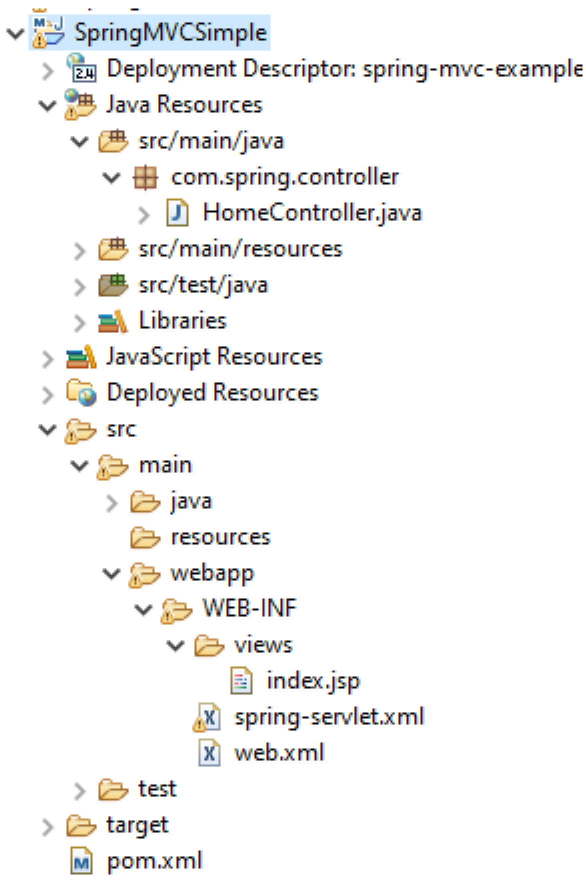
Now DispatcherServlet send the model object to the ViewResolver to get the actual view page.

Step 5:

Finally DispatcherServlet will pass the *Model* object to the *View* page to display the result.

Spring-MVC folder structure

Example-1



```
SpringMVCSimple
  Deployment Descriptor: spring-mvc-example
  Java Resources
    src/main/java
      com.spring.controller
        HomeController.java
    src/main/resources
    src/test/java
    Libraries
  JavaScript Resources
  Deployed Resources
  src
    main
      java
      resources
      webapp
        WEB-INF
          views
            index.jsp
          spring-servlet.xml
          web.xml
    test
  target
  pom.xml
```

index.jsp

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <h1>Welcome to first Spring MVC Application</h1>
</body>
</html>
```

spring-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans
xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd
        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">


    <annotation-driven />
    <context:component-scan base-package="com.spring" />
    <beans:bean

    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

</beans:beans>
```

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
id="WebApp_ID" version="3.0">
  <display-name>spring-mvc-example</display-name>

     <!-- Add Spring MVC DispatcherServlet as front
controller -->
     <servlet>
        <servlet-name>spring</servlet-name>
        <servlet-class>

org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
        <init-param>
              <param-name>contextConfigLocation</param-name>
              <param-value>/WEB-INF/spring-
servlet.xml</param-value>
         </init-param>
    </servlet>
     <servlet-mapping>
        <servlet-name>spring</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

 </web-app>
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
     <modelVersion>4.0.0</modelVersion>
     <groupId>spring</groupId>
     <artifactId>SpringMVCSimple</artifactId>
     <packaging>war</packaging>
     <version>0.0.1-SNAPSHOT</version>
     <name>SpringMVCSimple Maven Webapp</name>
     <url>http://maven.apache.org</url>
     <dependencies>
```

```xml
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>servlet-api</artifactId>
            <version>2.5</version>
            <scope>provided</scope>
        </dependency>

        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>4.3.9.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-web</artifactId>
            <version>4.3.9.RELEASE</version>
        </dependency>

        <dependency>
            <groupId>javax.servlet.jsp</groupId>
            <artifactId>jsp-api</artifactId>
            <version>2.1</version>
            <scope>provided</scope>
        </dependency>

    </dependencies>
    <build>
        <finalName>SpringMVCSimple</finalName>
    </build>
</project>
```

HomeController.java

```
package com.spring.controller;


import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

@Controller

public class HomeController {


   @RequestMapping("/")

    public String display()

    {

        return "index";

    }

}
```

Output-

Welcome to first Spring-MVC Application