We need to create web.xml file in that we have to configure DispatcherServlet entry and ServletName-servlet.xml

And another is controller that will have business logic. To create controller we are going to use here @controller annotation.

Basically **@Component** is using general purpose for creating bean any class in the spring factory it has following three types

\* **@Controller** it using for controllers in the web application<br/>

\* **@Service** it using for services for any application<br/>

\* **@Repository** it is using for back end services like data access object and repositories for accessing data from database.

**package** com.doj.spring.web.controller;

**import** org.springframework.stereotype.Controller;

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

/\*\*

\*

\* **@author** sumit saini

\* **@Component** is using general purpose for creating bean any class in the spring factory it has following three types<br/>

\* **@Controller** it using for controllers in the web application<br/>

\* **@Service** it using for services for any application<br/>

\* **@Repository** it is using for back end services like data access object and repositories for accessing data from database.

\*/

@Controller

**public** **class** SpringHelloController {

@RequestMapping("/")

**public** String sayHello(){

**return** "hello";

}

@RequestMapping("/welcome")

**public** String sayWelcome(){

**return** "hello";

}

}

**Hello.jsp**

<h1>Welcome to DOJ Classes for Spring MVC!!!</h1>

**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"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* 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>SpringWebHelloWorld</display-name>

<servlet>

<servlet-name>springweb</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<!-- <init-param>

<param-name></param-name>

<param-value>/WEB-INF/xyz-servlet.xml</param-value>

</init-param> -->

</servlet>

<servlet-mapping>

<servlet-name>springweb</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**springweb-servlet.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: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/context http://www.springframework.org/schema/context/spring-context.xsd"*>

<!-- Component scaning from this base package -->

<context:component-scan base-package=*"com.doj.spring.web.controller"*/>

<context:annotation-config/>

<bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/view/"*></property>

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

</beans>

**Explanation:-**

**Point1:-**

In Spring2.5 we have to use public ModelAndView methodName() method to show the view(i.e. **public** ModelAndView welcome(){

Map<String, String> model = **new** HashMap<>();

model.put("name", "Sumit");

**return** **new** ModelAndView("home","model", model);

}

) but now in above versions we can also use String to show view.

In Spring2.5 and below versions one controller was able to handle one request and we have to use ModelAndView return type in controller.

In Spring2.5 one controller is being used only for one purpose but in above versions one controller can use for multiple purpose.it is able to handle multiple request.

Also if you want you can write one controller for whole application.

**Point2:-**

How DispatcherServlet search for mapping file it search file which has name DispatcherServletName-servlet.xml like here DispatcherServlet name is springweb so mapping file is springweb-servlet.xml check in same Directory.

If you want to change directory of springweb-servlet.xml file then can change by using this:

<init-param>

<param-name></param-name>

<param-value>/WEB-INF/xyz-servlet.xml</param-value>

</init-param>

**Point3:-**

When we use @Controller annotation at that time it is also create SimpleUrlHandlerMapping so no need to extend any class to create controller just use @Controller annotation.

If you use @Component annotation instead of @Controller then it will through error no Handler Mapping found.