

Shiva Online Trainer : Mobile No : 9985538462

Stepst To Develop Spring Mvc Based Application

step 1: Create A Project add The maven dependencies

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

Step 2: Configure Dispatcher Servlet In Web.xml

```
<servlet>  
    <servlet-name>ds</servlet-name>  
    <servlet-class>  
        org.springframework.web.servlet.DispatcherServlet  
    </servlet-class>
```

```
<load-on-startup>1</load-on-startup>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
<servlet-name>ds</servlet-name>
```

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

```
</servlet-mapping>
```

Note: SpringConfiguration File Naming convention followed by
<servletname-servlet>.xml

Step 3: Create Spring Configuration File based Servlet Logical Name

DispatcherServlet, the framework will try to load the applicationContext from a file named [servlet-name]-servlet.xml located in the application's WebContent/WEB-INF directory. In this case, our file will be ds-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"
       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-4.3.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd">
<context:component-scan base-package = "com.shiva.controller" />

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

```
<property name = "suffix" value = ".jsp" />
</bean>
```

```
</beans>
```

Note :

Following are some important points about HelloWorld-servlet.xml file –

- The [servlet-name]-servlet.xml file will be used to create the beans defined, overriding the definitions of any beans defined with the same name in the global scope.
- The <context:component-scan...> tag will be used to activate the Spring MVC annotation scanning capability, which allows to make use of annotations like @Controller and @RequestMapping, etc.
- The InternalResourceViewResolver will have rules defined to resolve the view names. As per the above-defined rule, a logical view named hello is delegated to a view implementation located at /WEB-INF/jsp/hello.jsp.

Note: If not Configured InternalResourceViewResolver it goes to rootDirectory to find out the view component.

Step:4 Create A view Component in side the webapp folder.(root directory)

Index.jsp

```
<html>
```

```
<body>
```

```
<h2>Welcome to SpringMvc First Application!</h2>
```

```
<form action="hello">
```

```
<input type="submit" value="Go" />
```

```
</form>
```

```
</body>
```

```
</html>
```

Step :5 Create A Controller Class It Act as Helper Class/HandlerClass/ControllerClass

```
package com.shiva.controller;
```

```
import org.springframework.stereotype.Controller;
```

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

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
@Controller
```

```
public class WelcomeController {
```

```
    @RequestMapping("/hello")
```

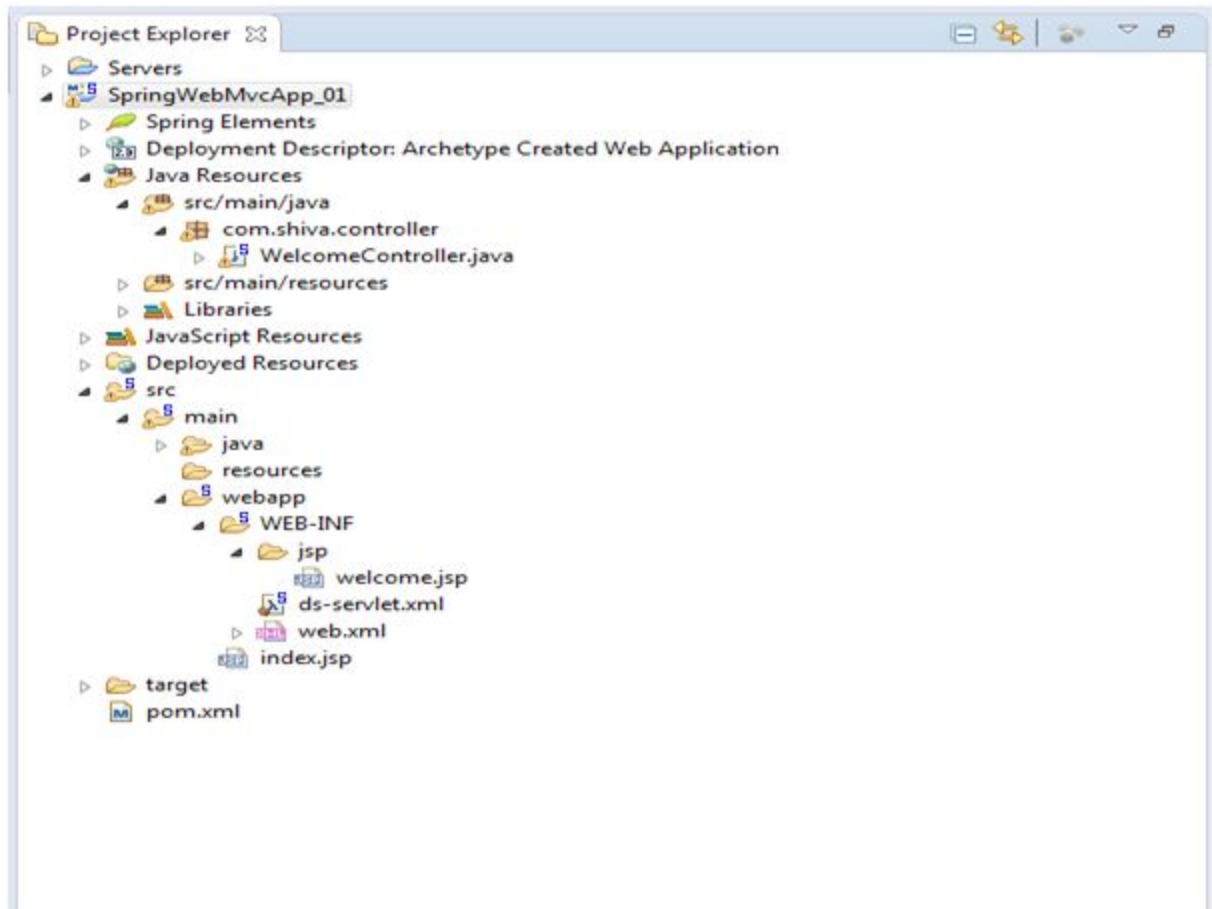
```
    public String sayHello()
```

```
    {
```

```
        return "welcome";
```

```
    }
```

```
}
```



Note: Controller class giving response back view or model Object to DispatcherServlet
DispatcherServlet receives Model or view some time it can receive ModelAndView(view and object also)it gives ViewResolver
ViewResolver takes the request from dispatcher Servlet it gives appropriate view component
If ViewResolver not configuredThen it gives to root Directory