**Spring MVC:**

**-----------**

**A Spring MVC is a Java framework which is used to build web applications.**

**It follows the Model-View-Controller(MVC) design pattern.**

**It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.**

**A Spring MVC provides an elegant solution to use MVC in spring framework by the help of DispatcherServlet(Front Controller)--web.xml.**

**Here, DispatcherServlet is a class that receives the incoming request and maps it to the right resource**

**such as controllers, models, and views.**

**Advantages of Spring MVC Framework**

**----------------------------------**

**Let's see some of the advantages of Spring MVC Framework:-**

**Separate roles - The Spring MVC separates each role, where the model object, controller, command object,**

**view resolver, DispatcherServlet, validator, etc. can be fulfilled by a specialized object.**

**Light-weight - It uses light-weight servlet container to develop and deploy your application.**

**Powerful Configuration - It provides a robust configuration for both framework and application classes that**

**includes easy referencing across contexts, such as from web controllers to business objects and validators.**

**Rapid development - The Spring MVC facilitates fast and parallel development.**

**Reusable business code - Instead of creating new objects, it allows us to use the existing business objects.**

**Easy to test - In Spring, generally we create JavaBeans classes that enable you to inject test data using the setter methods.**

**Flexible Mapping - It provides the specific annotations that easily redirect the page.**

**To Develop Spring Webapplication Using MVC and FC Pattern files are Written in below order**

**1)XMLFILE**

**a.web.xml :- to Configure FC(DispatcherServlet).**

**b.Spring Configuration Files : Activate Annotation and Provide View Resolvers**

**2)JavaFiles : Controller class with request method**

**3)UI Files : jsp/Html/css and JavaScript**

**a)web.xml**

**==========**

**<servlet>**

**<servlet-name>sampleFC</servlet-name>**

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

**</servlet>**

**<servlet-mapping>**

**<servlet-name>sampleFc</servlet-name>**

**<url-pattern>/mvc/\*</url-pattern>**

**</servlet-mapping>**

**b)Spring Config Files(xml)**

**==>it must be created under WEB-INF Location file name should follow naming convension**

**[servlet-name]-servlet.xml ==>sampleFc-servlet.xml**

**Ex:**

**----**

**sampleFc-servlet.xml**

**---------------------**

**Code for activatiing the annotations(@Controller)**

**<conext:component-scan base-package="com.ust"/>**

**Configuring the ViewResolver**

**Here InternalResourceViewResolver is a class which can be used to provide prefix(location) and suffix(extension) of Ui Files**

**Ex;**

**---**

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

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

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

**</bean>**

**Note:**

**-------------------**

**To Work SpringWebMvc we need to Configure spring-webmvc dependency in pom.xml**

**MVC + FC(control flow of spring mvc)**

=>It is a combination design pattern used in dynamic web application development here Front Controller is a servlet.

It will received request (HttpServlet) and dispatched to one controller (class) based on URL.

=>Controller execute logic and it may communicate to DB and fetch data which will be stored in model memory.

=> This model memory will be shared with view (Display Code).

=> Data will be placed in views file (Data Rendering) and finally view returned back to FrontController.

=> Front Controller sent this as HttpResponse



SPRING WEB-MVC AND FC DESIGN



1) FrontController is a servlet which behaves like entry and exit gate to application running in servlet.

2) In Spring FC is a pre-define servlet named as DispatcherServlet.

package : org.springframework.web.servlet

3) FC will read HttpRequest (URL) and identifies one controller method based on HandlerMapper (MAP).

4) HandlerMapper will be created by FC at runtime it is a map holds URL (key) and connected controller method (value).

5) Controller is a class (Spring Bean) which can have multiple methods (known as request method).

6) Controller method returns finally ModelAndView class object. It holds ViewName(required) and Model (data) (optional)

7) FC reads ModelAndView (MAV) object and call ViewResolver class to identify UI(view page).

8) ViewResolver contains prefix (location of view page) and suffix (extension of view page).

This is use to make controller independent of UI technology.

View Page = Prefix + view name + suffix

= /myfile/ home .jsp

9) ViewResolver finds view page nad view page reads data form model memory if exist using JSTL and EL (Data Rendering).

10)Finally view page as returned to FC and save returned as HttpResponse to client.

**NOTE:**

a) FC (Dispatcher Servlet) must be configured in web.xml file using directory match url pattern

Ex: /mvc/\*

b) HandlerMapper is auto created object.

c) Controller class is a spring bean

no of modules = no of controller in project

d) ViewResolver are pre-defined in spring must be configured either using XML or using java configuration.

e) JSTL (Jsp Standard Tag Library) used to write java code in tag based format to apply CSS/JS easily and for flexible output.