# Spring MVC Fundamentals

1. What is a Spring MVC ?

Ans -

* Spring MVC is a web framework built around principles of Spring for development of web application.
* Its POJO based and interface driven.
* Based on dispatcher servlet / Front controller pattern
  + MVC stands for Model-View-Controller
* Very lightweight and un-obtrusive framework
* Support for
  + Themes
  + Locales i18n
  + Restful Services
  + Annotation based configuration
  + Seamless integration with other spring services , beans. Whole dependency injection thing works very well here in Spring MVC

2. Below is the common vocabulary which is used in the Spring Web application project.

* DispatcherServlet - The entry /configuration point of application.
* Controller - Command pattern object that handles the request and determines which routes route to
* RequestMapping - The url and request type a method is tied to
* ViewResolver - Used to locate JSP pages or whatever view we are using
* Servlet-Config - Configuration file per dispatcher servlet
* POJO - Plain old Java Object
* Bean - A Spring configured POJO

Q - What application we are making using Spring MVC framework ?

Ans - We are basically making an web application which will be keeping record of exercise done by us.

Important points to note :-

* Spring Webflow project is built on top of Spring MVC.
* Below are the three dependencies from Maven point of view for building Spring MVC based application
  + spring-webmvc
  + servlet-api
  + jstl

So basically when we have to provide details of above JARS then we have to provide "groupId", "artifactId" and "version" of above jars in the pom.xml file and then maven will automatically download these above JARS and also any other transitive dependencies.

Q - What is "m2e" plugin ?

Ans - "m2e" is a popular eclipse IDE plugin with stated project goal "to provide a first-class Apache Maven support in the Eclipse IDE".

File - New - Other - Maven - Maven Project. "New Maven Project dialog box will come and here we have to choose archtype for the new project. To choose most appropriate type lets type "web" in the fliter and from there we can go ahead and create project.

Q - What is meaning of "scope=provided" ? This is in context of when we add dependencies in our pom.xml file .

A - "scope=provided" means - do not package this JAR with the WAR file of the application. This means that this JAR will be provided by our application server at runtime.

Q - What are the main points of Spring MVC Configuration ?

Ans - Below are the main points of Spring MVC Configuration.

* Configure web.xml
* Configure servlet-config.xml
* Add a controller
* Add a view

Q - What application server we are using for this spring mvc application ?

Ans - We are using Tomcat for this development. Please note that a standalone Tomcat server is just a web container and it does not provide some of other Java EE features which are provided by other App servers in market like WebSphere, JBoss etc. However this should be enough for our development for the demo of spring mvc application.

Look at the servlet specification and copy schema definition for the same in your web.xml file. Just remember that typing schema definition in the web.xml file can be error prone and we have to be very careful while typing that.

Q - What is "notion of" namespaces while configuring "Bean Definition File" ?

Ans - Namespace have lots of defined functionality with them. This is just the example of using them in our spring bean definition file. We have seen lots of time clumsy definition of namespace in a typical spring definition file. So basically we define namespace and then we use them. Below is the example of defining a namespace and then using it,

<beans xmlns="http://....."

xmlns:mvc="http://...."

xmlns:context="http://.."

</beans>

Below is the example of its usage ,

<mvc:annotation-driven/>

<context:component-scan base-package="com.pluralsight.controller"/>

So basically you define a namespace and then basically you define a tag in our code and we are done. Below is the URL of local workspace which is working now,

<http://localhost:8080/FitnessTracker_Spring_MVC/greeting.html> . So now that project setup for sample Spring MVC application is done a running fine, lets recap the settings / configuration which we have done,

* We have done configuration in web.xml file. Basically we setup config for Dispatcher Servlet in this file. Class name of Dispatcher servlet is org.springframework.web.servlet.DispatcherServlet . We also provided location of context file needed for initialization of DispatcherServlet. Below is sample configuration snippet taken from workspace,

<servlet>

<servlet-name>fitTrackerServlet</servlet-name>

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

<init-param>

<param-name>contextConfigLocation</param-name>

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

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>fitTrackerServlet</servlet-name>

<url-pattern>\*.html</url-pattern>

</servlet-mapping>

And we also did configuration in servlet-config.xml file. Below is the sample,

<?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"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.2.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-3.2.xsd"*>

<mvc:annotation-driven/>

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

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

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

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

</bean>

<!--

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

p:prefix="/WEB-INF/jsp/" p:suffix=".jsp" />

-->

</beans>

Q. What are diffferent return types of methods in spring mvc which are annotated with @RequestMapping annotation ?

Ans - Below are the diffferent return types ,

* ModelAndView
* Model
* Map
* View
* String

JSTL and EL are expression language of JSP. These expression language can be used to write code in JSP.

Q. How does request routing happens ?

Ans - Refer to the below mapping in web.xml file ,

<servlet-mapping>

<servlet-name>fitTrackerServlet</servlet-name>

<url-pattern>\*.html</url-pattern>

</servlet-mapping>

Above mapping says that anything (i.e any request) comes with having \*.html as url pattern, please forward it to fitTrackerServlet to handle it. We can look at the detailed configuration of fitTrackerServlet, which is

<servlet>

<servlet-name>fitTrackerServlet</servlet-name>

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

<init-param>

<param-name>contextConfigLocation</param-name>

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

</init-param>

</servlet>

So we see above that DispatcherServlet is configured using servlet-config.xml. And inside servlet-config.xml we have configuration,

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

So it will scan all the class having annotation @Controller

Q - What is a @ModelAttribute tag ?

Ans - @ModelAttribute is an annotation that binds a method parameter or return value to a named model atttribute and then exposes it to a web view.