Spring MVC

** Model 1 and Model 2 (MVC) Architecture

Before developing any web application, we need to have idea about design models. There are two types of programming models (design models)
Model 1 Architecture
Model 2 (MVC) Architecture

Model 1 Architecture
Servlet and JSP are the main technologies to develop the web applications.

Servlet is considered faster alternative to CGI. Servlet technology doesn't create process, rather it uses threads from a thread pool to handle request. The advantage of creating thread over process is that it doesn't allocate separate memory area , leading to better performance. Thus many subsequent requests can be easily handled by servlet.

Problem in Servlet technology Servlet needs to recompile if any designing code is modified. It doesn't provide separation of concern. Presentation and Business logic are mixed up.

JSP overcomes almost all the problems of Servlet. It provides better separation of concern, now presentation and business logic can be easily separated. You don't need to redeploy the application if JSP page is modified. JSP provides support to develop web application using JavaBean, custom tags and JSTL so that we can put the business logic separate from our JSP that will be easier to test and debug.

Advantage of Model 1 Architecture
Easy and Quick to develop web application

Disadvantage of Model 1 Architecture
Navigation control is decentralized since every page contains the logic to determine the next page.
If JSPs names are modified, we need to change it in all the pages leadining to the maintenance problem.
Time consuming.
Hard to extend It is better for small applications but not for large applications.

Model 2 (MVC) Architecture
Model 2 is based on the MVC (Model View Controller) design pattern. The MVC design pattern consists of three modules model, view and controller.

Model The model represents the state (data) and business logic of the application. (JavaBean , POJOs)

View The view module is responsible to display data i.e. it represents the presentation.(JSP)

Controller : The Front controller module(Servlet/Filter) acts as an interface between view and model. It intercepts all the requests i.e. receives input and

sends commands to Model / View to change accordingly.

Advantage of Model 2 (MVC) Architecture

Navigation control is centralized .
Now only controller contains the logic to determine the next page.
Easy to maintain
Easy to extend
Easy to test
Better separation of concerns

Disadvantage of Model 2 (MVC) Architecture
Developing centralized dispatcher (a navigation controller) is tedius, especially when web app size grows. That's the reason , Spring MVC is so popular , which implements Servlet based MVC pattern n supplies readymade components : Front Controller , Handler mapping , View Resolver ...

What is MVC ?
Model-View-Controller --Standard design pattern , meant for separation of concerns(=tasks=responsibilities)

Model -- Java Bean (conversational state holder + B.L supplier) & POJOs

View layer --JSP , Thymeleaf/velocity/Angular/React/Vue
Represents UI / presentation logic (processing requests & generating response)

Controller -- Typically a servlet(used in Spring MVC) or a filter(used in Struts 2 framework)
Manages navigation & beans.

Front Controller -- A design pattern -- which ensures ANY request coming from ANY client , for this web app , will be intercepted by a common gate keeper(or a centralized dispatcher)
It will dispatch clnt request to further components , based upon nature of the req.

Which are the important blocks of Spring MVC ?

1. o.s.w.s.DispatcherServlet
Entry point of the spring MVC. Front Controller pattern
Centralized dispatcher
Will be intercepting all reqs coming from all clnts.
Managed by WC
web.xml . Life cycle started at the deployment time .
init ---> job of the D.S --> hybrid --xml
Def Name of master config file to start SC : spring-servlet.xml
Def loc : WEB-INF

read by D.S (DispatcherServlet)
API of SC : o.s.b.f.BeanFactory <---- o.s.c.ApplicationContext : for Java SE : in hybrid approach <---o.s.c.s.ClasspathXmlApplcationContext : instance , to start SC

In web app : API of SC : o.s.b.f.BeanFactory <---- o.s.c.ApplicationContext <---- WebApplicationContext<----- XmlWebApplicationContext : it's instance created by D.S @ web app dep time.

---

2. Request Handling Controller (Handler)
: prog supplied
Mandatory annotations :
@Controller : cls level annotation
@RequestMapping : method level annotation
which all HTTP requests , it can intercept ? : GET , POST , PUT , DELETE .....
@GetMapping : get
@PostMapping : post
@PutMapping : put
@DeleteMapping : delete

---

3. HandlerMapping Bean
auto populated by SC : by looking at req handling methods (@RequestMapping) present in req handling controller(@Controller)
Map
key : value of @ReqMapping(or it's sub type) annotation eg : /hello
value : F.Q Handler class Name+ Method name

---

4. View Resolver :
Job : Translation from Logical view (forward view name) ---> Actual view name
propeties : prefix , suffix + viewClass : JstlView (to enable JSTL tags/actions)
AVN(actual view name) = prefix + LVN + suffix
eg : /WEB-INF/views+LVN+.jsp

---

5. View Layer (JSP) : prog supplied

---

6. What is a model attribute?
It's the attribute(server side entry=k n value pair : String, Object)
Purpose : to store the results of B.L
Who creates --- Req handling Controller(handler) --prog supplied
Who sends it to whom : Handler ---> D.S
After D.S gets actual view name from V.R :
D.S chks : are there any model attrs :
Yes : D.S saves model attrs under Request scope & then forwards the clnt to view layer .
NO : D.S forwards the clnt to view layer .
How to access these model attrs from JSP ?

${requestScope.attrName}

** What are different ways for handler to add model attrs ?

6.1 Via   o.s.w.s.ModelAndView?

o.s.w.s.ModelAndView : class => holder for model attrs + logical view name
Ctor :
ModelAndView(String viewName,String modelAttrName,Object modelAttrVal)
eg : what can be valid ret type of req handling method
String OR ModelAndView : ModelAndView

6.2 Any Simpler way to send model attributes from Handler --> D.S ?

Use o.s.ui.Model : i/f ---holder (Map) of model attributes
How do u add model attributes ?
public Model addAttribute(String modelAttrName,Object modelAttrVal)
eg : How to add 2 model attrs? : method chaining

Who will supply empty Model map (as the dependency)? : SC
IoC : DI
How to tell SC that handler method needs a model map ? : add it as the arg of req handling method
When req handling controller rets string : logical view name (Handler implicitly rets all model attr map to D.S)
---

7. Handling request parameters in Controller ?
Request Handling method argument levele annotation
@RequestParm
To bind request paramters to the method arguments

eg : @RequestParam("price") double productPrice
SC : double productPrice =Double.parseDouble(request.getParameter("price"))

Reco : Match req param names with method arg names.

eg : URL :
http://localhost:8080/day13.1/test/product?nm=pen&qty=10&price=40.5&manuDate=2020-1-1

Problem : HTTP 400 :
Bad request => some request data coming from client is invalid
Default date format : MM/dd/yyyy
To tell SC : use @DateTimeFormat annotation , along with @RequestParam
---