# Spring MVC/ spring boot MVC Flow

Spring Boot MVC = spring MVC ++

=>Spring MVC/Spring boot MVC is designed around a pre-defined FrontController servlet called DispatcherServlet i.e all operations and flows in the web application takes place under the control of DispatcherServlet.

=> The main comps of spring MVC/Spring boot MVC app are
- a) DipsatcherServlet [ Front Controller ]
- b) Handler /Controller classes [Java classes having request delegatation logics to service,DAO classes]
- c) HandlerMapping [To map/link DispatcherServlet trapped requests to Handler/Controller classes)
- d) View comps (Physical View comps of any technology like jsp having presentation logic to format the results)
- e) ViewResolver ( To resolve/indentity Physical view comp of different Technologies based on the given Logical View names]

note: HandlerMapping is helper comp that indentifies correct handler /controller class to handle the request trapped by DispatcherServlet (DS)
note: ViewResolver is helper comp that resolves/indentifies correct physical view comp name and location to format results by executing the presentation logics.

conclusion :: To send the DS trapped request to appropriate handler method of handler class we need the support of Handler Mapping component
To send and format the Handler method generated /gathered results by indentifying the appropriate physical view comp (like jsp) , we need ViewResolver comp

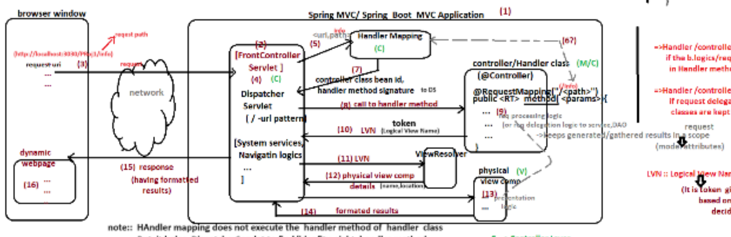
=> To cfg java class as spring bean cum controller/handler class , we need @Controller annotation

=> To cfg java method of controller class /handler class as the handler method with request path we can use @RequestMapping annotation

note:: one handler class can have multiple handler methods with unique request paths

=> Servlet comp is not identified with its class name , but it is identified with its url pattern
=> spring bean is not identified with its class name ,but it is identified with its bean id
=> Controller class handler method is identified with its request path that is given in @RequestMapping(-)

```
@Controller
public class MyController {
                          |--->request path
 @RequestMapping("/welcome")
public   String showHomPage(){
 ...
 ..              Every Handler method of the controller/handler class
 }               is identified with its   request path
}
```


browser window
Spring MVC/ Spring Boot MVC Application
(1)
request path
(http://localhost:8080/Proj1/info)
request uri
(3)
request
network
(2)
[FrontController Servlet ]
(4) (C)
Dispatcher Servlet
( / -url pattern)
[System services, Navigatin logics
...
]
<url,path> info
(5)
Handler Mapping
(C)
(6?)
(7)
controller class bean id, handler method signature to DS
controller/Handler class (M/C)
(@Controller)
(/info)
@RequestMapping("/<path>")
public <RT> method( <params>){
(8) call to handler method
(9) req processing logic (or req delegation logic to service,DAO
->keeps generated/gathered results in a scope
request
(model attributes)
token
(10) LVN (Logical View Name)
(11) LVN
ViewResolver
(12) physical view comp details (name,location)
physical (V) view comp
(13)
presentation logic
(14) formated results
dynamic webpage
(16)
(15) response (having formatted results)

note:: HAndler mapping does not execute the handler method of handler class
But it helps DispatcherServlet to find/idenfity right handler method of handler class to process the request .DispatcherServlet directly calls the handler method

C---> Controller Layer
V ---> View Layer
M ---> Model Layer

=>Handler /controller class comes under Model Layer
if the b.logics/request processing logics are kept directly in Handler methods

=>Handler /controller class comes under Controller layer
If request delegation logics to interact with service/DAO classes are kept in handler methods

LVN :: Logical View Name
(It is token given to ViewResolver comp through DS based on which the name and location of the physical view comp will be decided)

(1) Programmer deploys the spring MVC/spring Boot MVC web application in web server or applicationm server

(2) Becoz the of <load-on-startup> enabled on DispatcherServlet (DS) the DS will be pre-instantiated and initialized.. As part of DS initialization the IOC Container will be created of type ApplicationContext.. This IOC container performs pre-instantation of singleton scope spring beans like HandlerMAppings, Handler classes, service, DAO classes, View Resolvers and etc.. also completes the neccessary dependancy injections and keeps the spring beans in the ineternal cache of IOC container.

(3) Browser gives the request to spring MVC /spring boot MVC web aplication

(4) As front controller DS traps and takes the request and applies the common system services

(5) DS hand overs the trapped request to Handler Mapping comp

(6?) HandlerMapping comp seraches in all @Controller classes for the handler method whose request path matches with the path of currently trapped request ( uses reflections api for this)

(7) HandlerMapping comp gives that matched handler Method sinnature and Controller class bean id to DS

(8) DS submits the recieved bean id to DS mangaged IOC container and gets Controller class obj and calls the recived signature based handler method on controller object.

(9) The handler method of handler/controller class either directly process the request or delegates the request to service,DAO classes for request processing and also keeps the recieved results in Model attributes (mostly in request scope)

(10) The handler method of Handler class returns LVN (Logical View name) to DS

(11) DS gives LVN to ViewResolver

(12) ViewResolver resolves/indentifies the physcial view comp based on the recieved LVN .. ViewResolver gives the physical View comp details back to DS

(13) DS forwards /communicates with physical View comp where results are gathered from request scope and formatted using presentation logics

(14) DS gets the formatted results from the physical View comp

(15) DS sends the formatted results to browser as response

(16) browser displays the recieved response as dynamic web page

note:: View Resolver does not execute the view comp... It just helps the Dispatherservlet in the idenfication of Physical view comp like jsp file

note:: The process of deploying the web application or distributed app in web server or App server is called Deployment and the process of removing the app from web server or App server is called undeployment

Application server = web server++

Examples of web servers:: Tomcat upto6.x , Jetty , Jonus ,undertow , resin and etc..
Examples of Application servers :: weblogic , wildfly , jboss , jrun , tomcat 7.x+ , GlassFish and etc..

**Spring Boot MVC = spring MVC ++**

**Spring MVC/ spring boot MVC Flow**

===

**=>Spring MVC/Spring boot MVC is designed around a pre-defined FrontController servlet called DispatcherServlet i.e all operations and flows in the web application takes place under the control of DispatcherServlet.**

of

**=> The main comps spring MVC/Spring boot MVC app are**

**a) DipsatcherServlet [Front Controller]**

**b) Handler/Controller classes [Java classes having request delegatation logics to service,DAO classes]**

**) HandlerMapping [To map/link DispatcherServlet trapped requests to Handler/Controller classes)**

**/d) View comps (Physical View comps of any technology like jsp having presentation logic to format the results)**

**e) ViewResolver (To resolve/indentity Physical view comp of different Technologies based on the give Logical View names]**

**note: HandlerMapping is helper comp that indentifies correct handler /controller class to handle the request trapped by DispatcherServlet (DS)**

**note: ViewResolver is helper comp that resolves/indentifies correct physical view comp name and location**

**to format results by executing the presentation logics.**

**conclusion :: To send the DS trapped to request appropriate handler method of handler class we need the support of Handler Mapping component To send and format the Handler method generated /gathered results by indentifying the appropriate physical view comp (like jsp) , we need ViewResolver comp**

cum

**=> To cfg java class as spring bean controller/handler class, we need @Controller annotation**

**=> To cfg java method of controller class /handler class as the handler method with request path we can use @RequestMapping annotation note:: one handler class can have multiple handler methods with unique request paths**

**=> Servlet comp is not identified with its class name, but it is identified with its url pattern**

**=> spring bean is not identified with its class name,but it is identified with its bean id**

**=> Controller class handler method is identified with its request path that is given in @RequestMapping(-)**

**@Controller**

**public class MyController {**

**|--->request path**

**@RequestMapping("/welcome") public String showHomPage(){**

**Every Handler method of the controller/handler class**

**is identified with its request path**

**browser window**

**reqest path**

**(http://localhost:3030/PROj1/info) request uri**

(3)

requ

Spring MVC/ Spring Boot MVC Application

(1)

info

<uri,päth (5)

Handler Mapping

(2) [FrontController

Servlet ] (4) (C) Dispatcher Servlet

controller class bean id,

(8) call to handler method

(6?)

(C)

(7)

controller/Handler class (M/C) (@Controller)

Vinfo)

@RequestMapping("/<path>") public <RT> method('<params>{

(9)

req processing logic

(or reg delegation logic to service,DAO ->keeps generated/gathered

=>Handler /controller class comes under Model Layer if the b.logics/request processing logics are kept directly in Handler methods

=>Handler/controller class comes under Controller layer

if request delegation logics to interact with service/DAO classes are kept in handler methods

request

results in a scope

(mode attributes)

LVN :: Logical View Name

(It is token given to ViewResolver comp through DS based on which the name and location of the physical view comp will be decided)

network

handler method signature to DS

(/-url pattern

token

(10) LVN (Logical View Name)

dynamic webpage

(15) response

[System services, Navigatin logics

**ViewResolver**

(11) LVN

**physical (V)**

**(16)**

**(having formatted results)**

**(12) physical view comp**

**view comp**

**details (name,location)**

**(14)**

**formated results**

**note:: Handler mapping does not execute the handler method of handler class But it helps DispatcherServlet to find/idenfity right handler method**

(13) refentation

logic

**C---> Controller Layer**

**of handler class to process the request .DispatcherServlet directly calls the handler method V ---> View Layer M ---> Model Layer**

**(1) Programmer deploys the spring MVC/spring Boot MVC web application in web server or applicationm server (2) Becoz the of <load-on-startup> enabled on DispatcherServlet (DS) the DS will be pre-instantiated and initialized.. As part of DS Initialization the IOC Container will be created of type ApplicationContext.. This IOC container performs pre-instantation of singleton scope spring beans like HandlerMappings, Handler classes, service, DAO classes, View Resolvers and etc.. also completes the neccessary dependency Injections and keeps the spring beans in the ineternal cache of IOC container.**

**(3) Browser gives the request to spring MVC/spring boot MVC web aplication**

**(4) As front controller DS traps and takes the request and applies the common system services**

**(5) DS hand overs the trapped request to Handler Mapping comp**

**(6?) HandlerMapping comp seraches in all @Controller classes for the handler method whose request path matches with the path of currently trapped request (uses reflectiona api for this)**

**(7) HandlerMapping comp gives that matched handler Method sinnature and Controller class bean id to DS**

**(8) DS submits the recieved bean id to DS mangaged IOC container and gets Controller class obj and calls the recived signature based handler method on controller object.**

**(9) The handler method of handler/controller class either directly process the request or delegates the request to service, DAO classes for request processing and also keeps the recieved results in Model attributes (mostly in request scope)**

**(10) The handler method of Handler class returns LVN (Logical View name) to DS**

**(11) DS gives LVN to ViewResolver**

**note:: View Resolver does not execute the view comp... It just helps the Dispatherservlet in the idenfication of Physical view comp like jsp file**

**note:: The process of deploying the web application or distributed app in web server or App server is called Deployment and the process of removing the app from web server or App server is called undeployment**

**Application server = web server++**

**Examples of web servers:: Tomcat upto6.x, jetty, jonus,undertow, resin and etc..**

**Examples of Application servers :: weblogic, wildfly, jboss, jrun, tomcat 7.x+, GlassFish and etc..**

**(12) ViewResolver resolves/indentifies the physcial view comp based on the recieved LVN.. ViewResolver gives the physical View comp details back to DS**

**(13) DS forwards/communicates with physical View comp where results are gathered from request scope and formatted using presentation logics**

**(14) DS gets the formatted results from the physical View comp**

**(15) DS sends the formatted results to browser as response**

**(16) browser displays the recieved response as dynamic web page**