





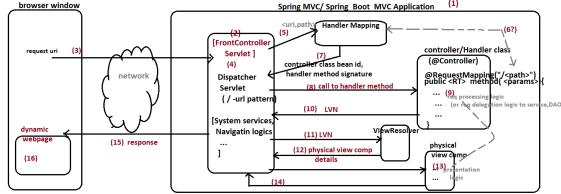
**Spring MVC/ Spring boot MVC Flow**  
=====

Spring Boot MVC = spring MVC ++

=> The main components in MVC/Spring MVC app are

- DispatcherServlet / FrontController [ ]
- Handler / Controller classes [ Java classes having request delegation logic to service/DAO classes]
- HandlerMapping [ To map/link DispatcherServlet trapped requests to Handler/Controller classes]
- ViewResolvers [ Physical View comps of any technology like jsp having presentation logic to format the results]
- ViewResolver [ To resolve/identify Physical view comp of different Technologies based on the give Logical View names]

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



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

(2) Becoz 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-initialization of singleton scope spring beans like HandlerMappings, Handler classes, service, DAO classes, View Resolvers, and etc. also creates the necessary dependency injections and keeps the spring beans in the internal cache of IOC container

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

(4) 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 searches in all @Controller classes for the handler method whose request path matches with the path of currently trapped request ( uses reflection api for this)

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

(8) DS submits the received bean id to DS managed IOC container and gets Controller class obj and calls the received 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 received 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/identifies the physical view comp based on the received 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 received response as dynamic web page



