**SPRING MVC FLOW**

1. **Checked for the deployment descriptor ( web.xml). Whenever we are making a request.**

**As it is having the mention for the FrontController.**

1. Each incoming request first hits the **Front Controller ( DispatcherServlet)**  , which is the heart of the application. The **Front Controller** dispatches the request to the handlers, and allows the developers to use different features of the framework.
2. Front Controller has its own **WebApplicationContext**, which is inherited from the **WebApplicationContext** root. The beans configured in the root application can be accessed and shared between the context and Servlet instance of the application. As applicable to all the Servlets , the Front Controller gets initialized on the first request.
3. Once the **Front Controller** is initialized , it further looks for an XML file named **servlet\_name-servlet.xml** under the **WEB-INF** folder. It contains MVC specific components.
4. This configuration file is , by default, named **XXX-servlet.xml** under the **WEB-INF** folder. This file contains the mapping information of the URL to the controllers(handler method in our example it was greet), which can handle the incoming request. We used annotation-based controllers/handlers.
5. The **RequestMappingHandlerMapping** interface searches all the controllers to look for the **@RequestMapping**/@GetMapping annotation under **@Controller**.
6. After scanning all the user-defined controller, the appropriate controller based on URL mapping is chosen, and the appropriate method is invoked. The method selection takes place based on the URL mapping and the HTTP method that it supported.

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

1. After the execution of business logic written in the controller method,, now it’s time to generate the response. This is different from our usual HttpResponse, as it won’t be served to the user directly . Instead , the response will be sent to the Front Controller. Here , the response contains the logical name of the view, the logical name of the model data, and actual data to bind.
2. The logical view name is with the Front Controller, but it doesn’t give any information about the actual view page to return to the user. The bean for ViewResolver configured in the **XXX-servlet.xml** file will be the mediator to map view name to the actual page. There is a wide range of view resolvers supported by the framework.
3. **ViewResolver** helps to get the actual view that the Front Controller can return as a response. The **Front Controller** will render it by extracting the values from the bounded model data, and will return it to the user.