Date : 12/02/2021
Spring Boot 9AM
Mr. RAGHU

Spring Boot : Web MVC

*) MVC is a design pattern used to implement web applications

M = Model (Data/Entity classes)
V = View (UI/HTML)
C = Controller (Request processing code/class)

*) Controller suports HTTP Protocol (request/response).
1 Project/Application - modules modules
1 Gmail App - Inbox, User Register/Login, Sent, Draft, Spam ..etc Modules
1 BusTicket(redBus) -- User, Search, BookTicket, Tracking, Cancel ...etc

1 Module--1 Controller
ex : UserController(C), AdminController(C), InboxController, ..etc

*) Java -Sun/Oracle says - 'Use Servlets API for web applications'.
***** At least one servlet is required to implemented webapp in java

-------------------HandlerMapping------------------------

Key(Path+ Http Method)	Value(Controller#method)
/emp/show (GET)	EmployeeController#showData()
/std/save (POST)	StudentController#saveData()

Q) Why ViewResolver?
A) It makes Controller code independent of UI technology.
Tomorrow after writing code, if we want to modify/upgrade
to new UI technology, no code changes required in Controller
if we use View Resolver.

-------Execution Flow---

#1. Browser makes request to server in 3 different ways.

a. Enter URL in addressbar [GET] (http://facebook.com)
b. Click Hyper Links [GET] (<a> tags) (Logout, Inbox, Members, Setting...)
c. HTML FORM Submit [GET/POST] (Register, Login, Payment, Comment, Feeback, Enq..etc)

#2. Request is sent to Server
Http Request contains 2 parts = Head(Init Line) + Body

Http Protocol:
https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
Http Heeaders:
https://en.wikipedia.org/wiki/List_of_HTTP_header_fields

#3. Spring F/w (web MVC) has provided one Pre-Defined Servlet
named as : DispatcherServlet(C)

-> This is only and entry and exit point. ie
It will read request and finally it gives response back.

*) We define multiple controllers in application,
those controller details are stored in map format.

--sample controller code---

```
@Controller
@RequestMapping("/emp")
class  EmpController {
   @RequestMapping(value="/show",method=GET)
   public String showHome() {
      ...
   }
   @RequestMapping(value="/save",method=POST)
   public String saveEmp() {
      ...
   }
}
```

----

*) One HandlerMapping object is created at runtime with all above
controller details

------------------HandlerMapping------------------------

Key(Path+ Http Method)	Value(Controller#method)
/emp/show (GET)	EmpController#showHome()
/emp/save (POST)	EmpController#saveEmp()

Who is Writing Controller code ----?   Programmer
Who is creating Object for Controller---?   Spring Container
Who is executing/calling controller#method   ---? FrontController/FC
When Controller method is called ---? When browser makes request.

If browser made /emp/show 10 times --- then 10 times Controller#method is called.

#4. FC gives Path and Method Details to handlermapping

#5. Fc gets back Controller and method details.

#6. FC will call method by reading object from container like
Ex:

```
empController.showData()
```

(For 1 Request = 1 time method is called)

```
#7. Controller(is a class) contains request processing code(inside
methods)
    those are executed by FC.

-> Browser is sending Login data, validate it.
-> Browser is asking for Logout, [session invalidation]
-> Browser is asking to post one comment [ save data in db and refelect
            back to ui]
 ..etc

 Every Java method --(Controller#method) is connected
 with Path and Http Method.

 EmpController#showData()  : Path:/show, Http Method : GET

*) URL(Path) is case-sensitive : /show, /SHOW, /Show .. are different.
--------------------------------------------------------------------
```