Date : 15/02/2021
Spring Boot 9AM
Mr. RAGHU

Spring Boot: Web MVC (First Application with setup)

a) Use Spring web dependency that comes with Embedded Tomcat.
b) Default port number for every Spring Boot web app is 8080.
   if we want to modify use : server.port=____

c) FC, HandlerMapping, ViewResolver are configured by Spring Boot only.

d) In Spring , FC is configured using web.xml (Here not required).
    ViewResolver can be configured using spring XMLconfig (<bean>)
    or Java based Config (@Configuration and @Bean).
    In Boot not required to configure.

e)** But we need to provide prefix and suffix using application.properties

```
    spring.mvc.view.prefix = _________
    spring.mvc.view.suffix = _________
```

f) Rules to write controller (not giving all here)
-> Define one class with methods
-> Apply @Controller (@Component[creating object] + HTTP Protocol)
-> Every Controller method can have PATH (URL) and Http Method Type
   (GET, POST). So, apply

```
   @RequestMapping(value="/path",method=RequestMethod.GET)
   @RequestMapping(value="/path",method=RequestMethod.POST)

   --Short Format--
    @RequestMapping(value="/path") //default is GET.
```

*) Browsers supports only GET and POST. In General GET is used to
   fetch data from server and POST is used to send data to server.

-> GET Request never supports body, we can send only parameters
   using URL.

---Example Controller sample------------------

```
package in.nareshit.raghu;

@Controller
public class EmployeeController {

  //PATH + Http Method Type
  //@RequestMapping(value="/emp",method=RequestMethod.GET)
  @RequestMapping(value="/emp") //default is GET
  public String showEmpPage() {
    return  "EmployeeHome";
  }

}
```

```
*) Method returns String (viewName, No extension and No location).
*) RequestMapping annotation links URL,MethodType with
Controller#method
  ie if we enter http://localhost:8080/emp in addressbar (GET)
  then FC calls EmployeeController#showEmpPage()
---------------------------------------------------------
JSP[Java Server Pages] Views:-

-> When we add Spring web Dependency, Embedded Tomcat comes with only
   CATALINA  not with JASPER(ie no JSP support).

->  To work with JSP pages additionally we should add :
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>

======================code===================================
S#1 Create one Spring Boot Starter Project

Name : SpringBoot2WebMvcFirstApp
Dep  : Spring web

S#2 Add tomcat embedded jasper in pom.xml, to suport JSP pages.

<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
</dependency>

*)Note : Update Maven project (Alt+F5)
> Right click on Project > Maven > Update Project
> Choose checkbox [v] Forace Update of ...
> Click on OK.

S#3 Create folder as shown below
src
 |-main
    |- webapp
       |- WEB-INF
            |- mypages

> Right click on folder name > new > folder > Enter name > finish

S#4 Create JSP page under mypages folder
> right click on 'mypages' folder > new > other > Search JSP File
> select JSP File Option > next > Enter name: EmpHome.jsp > Finish

---EmpHome.jsp---
<html>
<body>
<h1>WELCOME TO SPRING BOOT + WEB MVC FIRST APP</h1>
</body>
</html>

S#5 properties file
server.port=9898
```

```
#First / symbol is root folder (webapp)
spring.mvc.view.prefix=/WEB-INF/mypages/
spring.mvc.view.suffix=.jsp
```

S#6.Controller code

```
package in.nareshit.raghu.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
public class EmployeeController {

        // method : public String <methodName> { return viewName }
        //@RequestMapping(value = "/emp") //GET

        @RequestMapping(value = "/emp",
                        method = RequestMethod.GET)
        public String showHome() {
                return "EmpHome";
        }
}
```

S#7. Run Starter class and Enter URL in addressbar

URL:

http://localhost:9898/emp
