**Day 10: 9 – Aug 2024 – Web Application**

*Improve controller layer*

*Controller generally is Servlet.*

*Struts MVC, JSF and spring MVC base upon Core Servlet.*

*Limitation of servlet.*

*Servlet life cycle ie init, service and destroy etc.*

*Get, post, put and delete http protocol specific methods.*

*public class MyServlet extends HttpServlet {*

*public void doGet(HttpServlet req, HttpServletResponse res) {*

*}*

*public void doPost(HttpServlet req, HttpServletResponse res) {*

*}*

*}*

*Limitation*

*In on servlet class we can’t write more than one doGet method as well as doPost m ethod.*

Some time if we want to want request and response object. But still that web container provide use both object.

doGet and doPost is not meaningful method.

all methods part of HttpServlet by default inherits.

In Spring MVC we need to create normal class and on that class write @Controller annotation.

@Controller

class MyController {

@RequestMapping(value=”hello”,method=RequestMethod.GET)

public ModelAndView sayHello() {

// do some logic

ModelAndView mav = new ModelAndView();

mav.addViewName(“display1.jsp”);

return mav;

}

@RequestMapping(value=”hi”,method=RequestMethod.GET)

public ModelAndView sayHi() {

// do some logic

ModelAndView mav = new ModelAndView();

mav.addViewName(“display2.jsp”);

return mav;

}

}

In spring mvc inside web.xml file we need to configure front controller class provided by spring mvc module. Spring MVC provided DispatcherServlet as a front controller.

Steps

1. Index.jsp is first page.
2. Which contains hello as url pattern using hyperlink
3. Once you click on hyperlink the request pass to web.xml file. DispatcherServlet receive any request because we mention url-pattern as /
4. Then it search xml file start with prefix as servlet-name ie dispatcher-servlet.xml.
5. Inside this file we enable set of annotation as @Controller, @Service, @Repository, @Component part of com root package.
6. Then spring container check the all class with method with annotation as @RequestMapping. If path match it invoke that method and re-direct the specific specifd which mention in ModelAndView class.

Same project

We need to add mysql connector dependency and spring jdbc

Pom.xml file

We need mysql connector, spring webmvc and jdbc dependency

In dispatcher-servlet.xml file you need to do the di for DriverManagerDataSource and JdbcTemplate

Then create Model layer

JavaBean class as Login with emailid and password, setter/getter method with @Component annotation

LoginService with @Service annotation which contains two business method signIn and signUp

LoginDao with @Repository annotation with which contains two method signin and signup with autowired for JdbcTemplate etc.

LoginController with @Controller annotation do di for LoginService and signin and signup functionality