Day 4 : 16 Sep 2024

Servlet : Servlet is normal java program which help to crate dynamic web page on server side.

Servlet interface

GenericServlet class

HttpServlet class

class MyServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse res) {

PrintWriter pw = res.getWriter();

Pw.println(“Welcome to My Web Application”);

}

Public void doPost(HttpServletRequest req, HttpServletResponse res) {

PrintWriter pw = res.getWriter();

String emailid = req.getParameter(“emailid”);

String password = req.getParameter(“password”);

// we can write jdbc code

If(emailid.equals(“akash@gmail.com”) && password.equals(“123”)) {

pw.println(“success”);

}else {

pw.println(“failure”);

}

}

doDelete(HttpServletRequest req, HttpServletResponse res) {

}

doPut(HttpServletRequest req, HttpServletResponse res) {

}

}

<http://localhost:8080/ProjectName/hello>

<http://localhost:8080/ProjectName/hi>

<http://localhost:8080/ProjectName/login>

@Controller this class behave like servlet.

class LoginController {

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

public ModelAndView sayHello() {

}

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

public ModelAndView sayHi() {

}

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

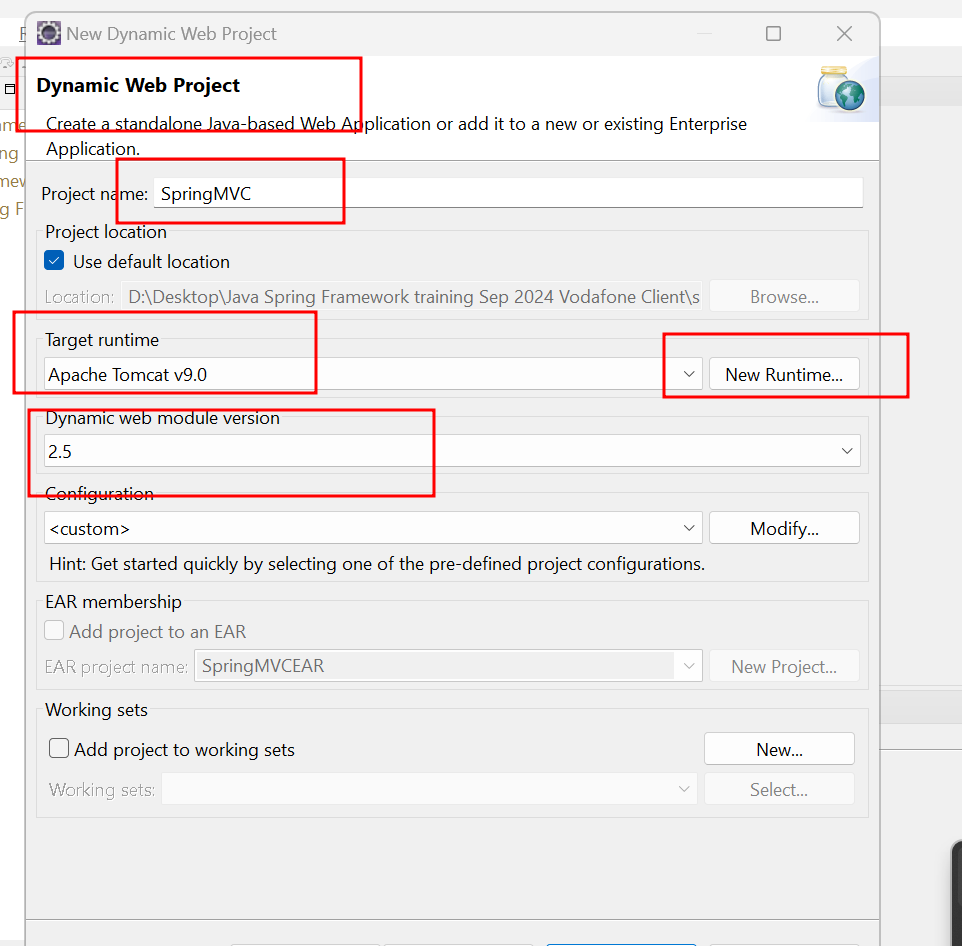
public ModelAndView login(HttpServletRequest req) {

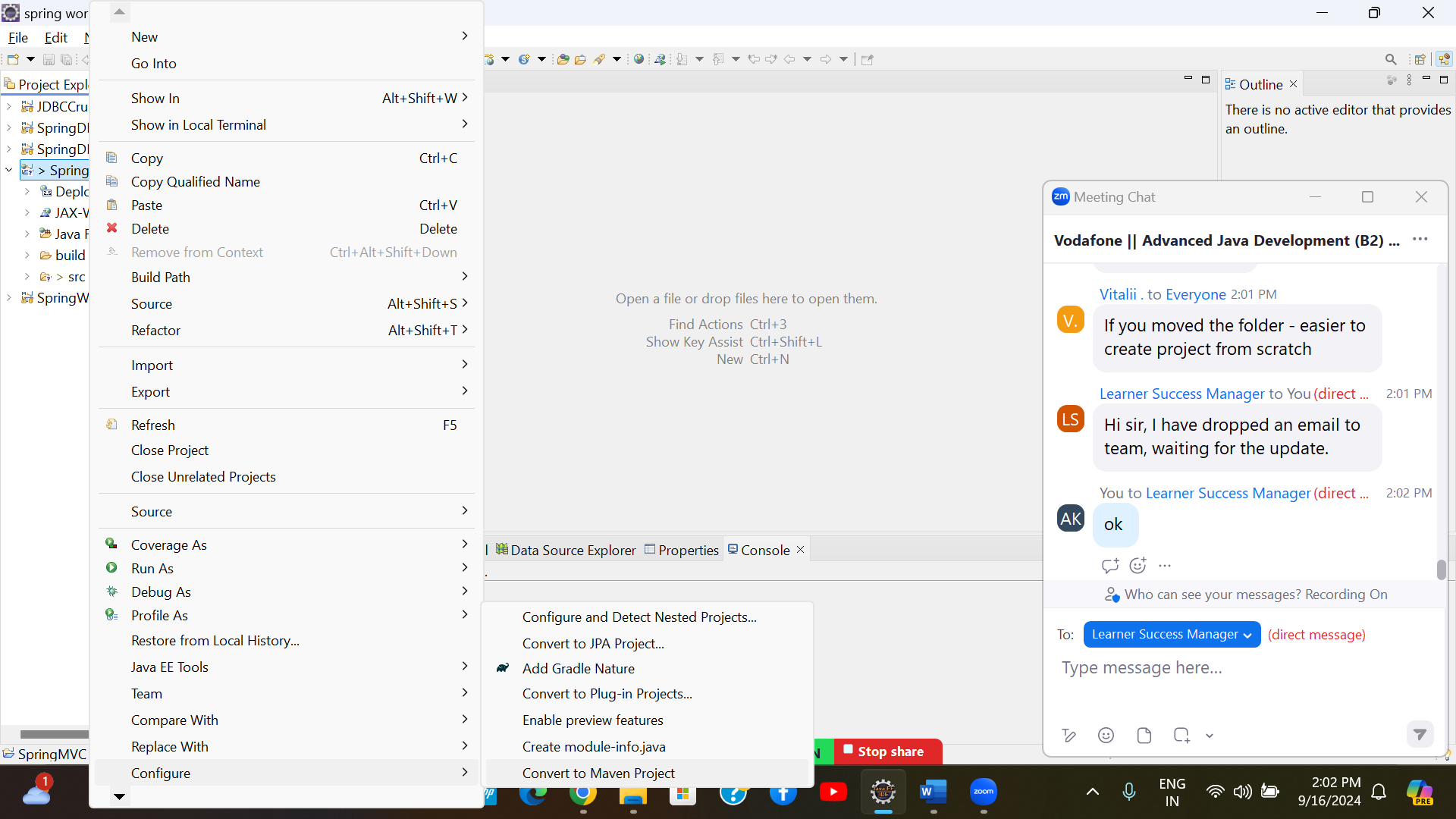
}

}

Tomcat open source server to develop the application

Creating dynamic web project in Eclipse IDE





Spring MVC internally follow MVC as well as front controller design pattern.

Front controller design pattern. It maintain all controller flow. They provided pre defined class ie DispatcherServlet. This class we need to configure inside web.xml file or class with @Configuration annotation.

Spring boot : Spring boot provide bootstrap for spring modules. If we develop any spring modules mainly spring MVC with spring boot more complex ie adding external server, adding more dependencies, adding configuration file using xml etc.

Spring boot = all spring modules – no xml file + few annotation + embedded internal web server ie tomcat.

Spring boot itself is core java or standalone production grade project (which contains main method in development mode) which help to create any type of project like web project.

Spring boot components

1. Spring boot starter : Spring boot provided lot of starter which help to all more than one jar file with help of only one dependencies.

We can develop spring boot using

1. Maven build tool pom.xml is part of maven tool not a part of spring boot.
2. Gradle build tool xml less they use build file to download the dependencies.

Few starter names as

Web starter, testing starter, jdbc starter, jpa (orm) starter, security starter, eureka server (micro service ) etc.

Spring boot auto configuration using this features spring boot automatically provide all configuration details base upon type of starter we added inside our project.

If we added web starter they provide web server features for core java project or standalone project.

To change port number, providing database information or providing some security features we need to use application.properties.

Spring boot provided one new annotation ie

@SpringBootApplication =@Configuation + @ComponentScan + @AutoConfiguration

Spring boot 3.x version

Spring boot 3.x version use spring modules 6.x version jar file and it need java min 17 version.