Spring MVC: Model View Controller

Before Spring framework or other framework if we want to develop any web application we were/are using Servlet and JSP to develop the web application.

If we want to create simple Servlet program we need to make normal java class and that class must be implements or extends type of servlet.

public class MyServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response) {

}

public void doPost(request, response) {

}

}

Spring MVC

@Controller

public class MyController {

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

public ModelAndView sayHello() {

// coding

ModelAndView mav = new ModelAndView();

mav.setViewName(“display1.html”);

return mav;

}

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

public ModelAndView sayHi() {

// coding

ModelAndView mav = new ModelAndView();

mav.setViewName(“display2.html”);

return mav;

}

}

In Spring MVC project View page ie html or jsp can’t call Controller method directly. We need configurate FrontController class provided by Spring framework in web.xml file or configuration class. This class consider as FronController means control all controller flow. Name of the class is DispatcherServlet.

Limitation of Spring Framework

1. Setting up dependencies
2. Database management
3. Linking files
4. Restart the server whenever we do any changes
5. To much configuration needed using xml
6. Redundant template code
7. Older architecture style
8. Dependency conflicts

**Spring boot :** Spring boot is another spring modules which help to bootstrap for all spring modules.

Spring boot is a standalone or core java project which help to create any type of project.

Spring boot = all spring modules – no xml file + few annotation + in build web server tomcat.

Spring boot components

Spring boot starter : Spring boot provided different types of starter which help to download all required dependencies base upon type of project we develop.

Web starter -🡪 it type of dependencies but combined all dependencies which help to develop the web application.

Jdbc starter -🡪

Jpa starter 🡪

Testing starter -🡪

Spring boot auto configuration : base upon type of starter we added in a project. Spring boot container provide all those resource for us or inject those resource like tomcat container if we add web starter, jdbc datasource if we jdbc starter.

Spring boot provided one the annotation

@SpringBootApplication : This annotation we need to write on main class.

This annotation is equal to

@SpringBootApplication = @Configuration + @AutoConfiguration +@ComponentScan

Inside main method we need to take the help of SpringApplication class which contains run method to launch spring boot application

SpringApplication.run(MainClass.class,args);

Spring boot +maven build tool : pom.xml file is a part of maven tool not a part of spring boot.

Spring boot + Gradle build tool : no xml file.