Day 3 : 12 Sep 2024

Spring JDBC or DAO module

Core and context module provided pre defined annotation

Ie

@Component : which we need to use on POJO class. generic annotation

@Repository : This annotation we need to use on class level ie DAO layer specific annotation.

@Service : this annotation we need to use on class level ie service layer specific.

@Autowired : property or complex property level.

We do the DI for DataSource API. DataSource is responsible to provide the database connection in secure manner. Before spring framework we were achieving Data Source features using server like web logic, jboss, web sphere etc. But spring jdbc module provided one the pre defined API ie DriverManagerDataSource which provide database connection. This configuration we can do using java classes with @Configuration annotation or xml file or properties file.

Spring JDBC module provided pre defined API ie JdbcTemplate. This class wrap core jdbc code and provided more method to improve jdbc code using spring jdbc modules.

Spring MVC ; Model View Controller

Using this module we can develop web application.

Before Spring or any other framework to develop server side programming language using Java technologies we are/were using servlet or jsp.

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) {

}

}