

[8] 폼데이터

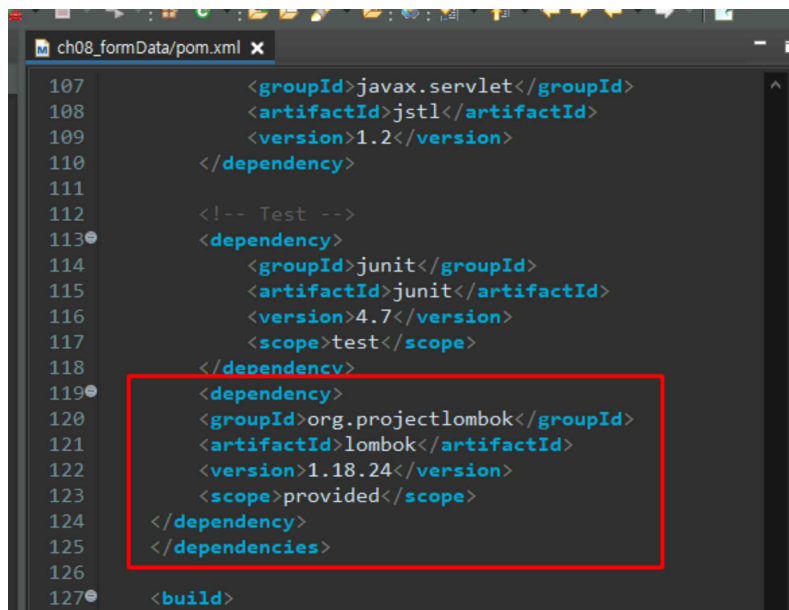
목표 : 데이터 입력을 받아 처리하는 기법들의 이해

HttpServletRequest, @RequestParam @ModelAttribute @PathVariable 등을 이용

우리가 jsp&Servlet을 할 때도 사용자가 form에 있는 데이터를 즉 input 태그나 select 태그를 이용해서 데이터를 전송을 했다. 그러면 서버에서는 그 데이터를 받아서 적합한 알맞은 로직을 수행후 다시 클라이언트에게 응답을 해주는 로직을 많이 구현해 보았다.

어차피 프로그램이라는 것은 데이터, 즉 사용자가 어떠한 요청을 하면 그 요청에 알맞은 데이터를 처리하고 그것에 대한 응답을 하는 게 거의 대부분의 프로그래밍으로 구성되어 있다.

Porm.xml 에 Lombok 추가

A screenshot of an IDE window titled 'ch08_formData/pom.xml'. The XML content shows a list of dependencies. A red rectangle highlights the addition of the Lombok dependency. The code is as follows:

```
107     <groupId>javax.servlet</groupId>
108     <artifactId>jstl</artifactId>
109     <version>1.2</version>
110 </dependency>
111
112 <!-- Test -->
113 <dependency>
114     <groupId>junit</groupId>
115     <artifactId>junit</artifactId>
116     <version>4.7</version>
117     <scope>test</scope>
118 </dependency>
119 <dependency>
120     <groupId>org.projectlombok</groupId>
121     <artifactId>lombok</artifactId>
122     <version>1.18.24</version>
123     <scope>provided</scope>
124 </dependency>
125 </dependencies>
126
127 <build>
```

Wep.xml 한글처리

```
ch08_formData/pom.xml  servlet-context.xml  web.xml x
6      <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
7      <context-param>
8          <param-name>contextConfigLocation</param-name>
9          <param-value>/WEB-INF/spring/root-context.xml</param-value>
10     </context-param>
11
12     <!-- Creates the Spring Container shared by all Servlets and Filters -->
13     <listener>
14         <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
15     </listener>
16
17     <!-- Processes application requests -->
18     <servlet>
19         <servlet-name>appServlet</servlet-name>
20         <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
21         <init-param>
22             <param-name>contextConfigLocation</param-name>
23             <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
24         </init-param>
25         <load-on-startup>1</load-on-startup>
26     </servlet>
27
28     <servlet-mapping>
29         <servlet-name>appServlet</servlet-name>
30         <url-pattern>/</url-pattern>
31     </servlet-mapping>
32
33     <!-- 한글처리 -->
34     <filter>
35         <filter-name>encodingFilter</filter-name>
36         <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
37         <init-param>
38             <param-name>encoding</param-name>
39             <param-value>UTF-8</param-value>
40         </init-param>
41         <init-param>
42             <param-name>forceEncoding</param-name>
43             <param-value>true</param-value>
44         </init-param>
45     </filter>
46     <filter-mapping>
47         <filter-name>encodingFilter</filter-name>
48         <url-pattern>/*</url-pattern>
49     </filter-mapping>
50 </web-app>
```

-join6

```
}
@RequestMapping("join6")
public String join6(@ModelAttribute("member") MemberDto memberDto) {
    return "member/result4";
    //ModelAttribute : Controoler 메소드의 파라미터나 리턴값을
    //Model 객체와 바인딩하기 위한 어노테이션
}
```

@ModelAttribute : Controoler 메소드의 파라미터나 리턴값을 Model 객체와 바인딩하기 위한

어노테이션

@ModelAttribute("member") MemberDto memberDto 값 넣어

return member/result4로 가