[ 08 ] 컨트롤러

1. 컨트롤러 클래스 제작
   1. 최초 클라이언트로부터 요청이 들어왔을 때, (DispatcherServlet이 매핑이 되어 있으므로, DispatcherServlet이 받고 HandlerMapping과 HandlerAdapter를 통해) 컨트롤러로 진입하게 되고, 컨트롤러는 요청에 대한 작업을 한 후 (ViewResolver를 통해 뷰를 찾아) 뷰 쪽으로 데이터를 전달한다. 그러면 뷰로 인해 화면에 응답을 하는 구조로 되어 있다.
   2. 컨트롤러 클래스 제작 순서
      1. @Controller 어노테이션을 클래스에 적용한다
         * 컨트롤러 역할의 클래스 파일은 servlet-context.xml에 <context:component-scan base-package=*"com.lec.ch08"* /> 명시된 패키지 아래에만 위치할 수 있다
      2. @requestMapping 어노테이션을 이용해서 요청 경로를 지정한다
      3. 웹브라우저의 요청을 처리할 메서드를 구현한다
      4. 뷰이름을 리턴한다
2. 요청 처리 메소드 제작.
   1. 클라이언트의 요청을 처리할 메소드를 제작한다
   2. 뷰페이지 이름 생성(조합) 방법

servlet-context.xml의 아래의 소스에 의해 조합된다.

<beans:bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<beans:property name=*"prefix"* value=*"/WEB-INF/views/"* />

<beans:property name=*"suffix"* value=*".jsp"* />

</beans:bean>

뷰페이지 이름 = prefix + 요청처리 메소드 반환값 +suffix

1. 뷰에 데이터 전달
   1. 컨트롤러에서 로직 수행 후 뷰페이지를 반환하고, 이 때 뷰에서 사용하게 될 데이터를 객체로 전달할 수 있다.

@RequestMapping("/board/write") //요청경로

**public** String content() {

**return** "/board/write"; // 뷰이름

}

* 1. Model 클래스를 이용한 데이터 전달 ; @RequestMapping어노테이션이 붙은 메소드는 Model형 객체를 매개변수로 받을 수 있다.
     1. Model 객체를 파라미터로 받음 ex. Public String content(Model model){
     2. Model 객체에 데이터를 담음 ex. model.addAttribute(“id”,”aa”);
     3. 컨트롤러에서 전달받은 Model 객체의 속성을 이용함 ex. ${id}

@RequestMapping("/board/content")

**public** String view(Model model) {

Member member = **new** Member("aaa","111");

// Model model = **new** BindingAwareModelMap();

model.addAttribute("id","aaa");

model.addAttribute("pw","111");

model.addAttribute("member", member);

**return** "/board/content";

}

* 1. ModelAndView 클래스를 이용한 데이터 전달 ; 매개변수로 데이터를 안 받고 매소드 안에서 직접 생성한다.
     1. ModelAndView 객체생성 ex. ModelAndView mv = new ModelAndView();
     2. Model 객체에 데이터를 담음 ex. mv.addObject(“id”,30);
     3. 뷰이름 설정 ex.mv.setViewName(“board/reply”);
     4. 컨트롤러에서 전달받은 객체의 속성을 이용함 ex. ${id}

@RequestMapping("/board/reply")

**public** ModelAndView reply(ModelAndView mav) {

mav.addObject("id","aaa");

Member member = **new** Member("aaa","111");

mav.addObject("member", member);

mav.setViewName("/board/reply");

**return** mav;

}

1. 클래스에 @requestMapping 적용
   1. 메소드마다 @RequestMapping 어노테이션을 적용하여 요청경로를 얻는다.
   2. 메소드에 @RequestMapping 어노테이션을 적용하여 요청경로를 얻는다. @RequestMapping 어노테이션을 클래스에 적용하여 요청경로를 얻는 방법도 있다.

// board/list

// board/content

// board/write

// board/reply

/\*클래스에 @RequestMapping적용하고 메소드에 @RequestMapping적용하면

\* 조합된 요청경로(클래스에 @RequestMapping적용한것+메소드에 적용한것)는 /board/reply로 한다.

\* @RequestMapping의 value나 반환된 뷰이름에 “/”는 유동적이다 \*/

@Controller

@RequestMapping("board") // 공통 요청 경로

**public** **class** BoardController {

@RequestMapping(value="write", method=RequestMethod.***GET***) // 요청경로 (class의 requestMapping + 메소드의 requetMepping = board/writeView) (board/writeView)

**public** String write() {

**return** "board/write"; // 뷰

// 뷰단 : "/WEB-INF/views/board/write.jsp"

}

@RequestMapping(value="content", method=RequestMethod.***GET***)

**public** String content(Model model/\*, HttpServletRequest request\*/) {

model.addAttribute("id", "aaa");

//request.setAttribute("pw", "111");

model.addAttribute("pw","111");

Member member = **new** Member("bbb","222");

model.addAttribute("member",member);

**return** "board/content";

}

@RequestMapping(value="list", method=RequestMethod.***GET***)

**public** ModelAndView list() {

ModelAndView mav = **new** ModelAndView();

Member member = **new** Member("ccc","333");

ArrayList<Member> list = **new** ArrayList<Member>();

list.add(member);

mav.addObject("list", list);

mav.setViewName("board/list");

**return** mav;

}

@RequestMapping(value="reply", method=RequestMethod.***GET***)

**public** ModelAndView reply(ModelAndView mav) {

Member member = **new** Member("ddd","444");

mav.addObject("member", member);

mav.setViewName("board/reply");

**return** mav;

}

}

**예제**

**pom.xml에 다음을 추가**

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.8</version>

<scope>provided</scope>

</dependency>

**web.xml에 다음을 추가(controller에서 add한 attribute값 한글 처리)**

(<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*pageEncoding=*"UTF-8"*%>은 페이지 내 한글처리)

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

**servlet-context.xml**

<annotation-driven />

<resources mapping=*"/resources/\*\*"* location=*"/resources/"* />

<resources mapping=*"/css/\*\*"* location=*"/css/"*/>

<beans:bean class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<beans:property name=*"prefix"* value=*"/WEB-INF/views/"* />

<beans:property name=*"suffix"* value=*".jsp"* />

</beans:bean>

<context:component-scan base-package=*"com.tj.ch09con.controller"* />

**MyController.java(새로운 컨트롤러 생성)**

**package** com.tj.ch08.controller;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

@Controller

**public** **class** MyController {

@RequestMapping("/")

**public** String main() {

**return** "main";

}

}

**MemberController.java**

@Controller

@RequestMapping("member")

**public** **class** MemberController {

@RequestMapping(params = "method=join")

**public** String join(Model model) {

model.addAttribute("kind", "회원");

**return** "member/join";

}

@RequestMapping(params = "method=login", method=RequestMethod.***GET***)

**public** String login() {

**return** "member/login";

}

@RequestMapping(params="method=logout", method=RequestMethod.***GET***)

**public** String logout() {

**return** "member/logout";

}

@RequestMapping(params="method=infoModify", method=RequestMethod.***GET***)

**public** String infoModify() {

**return** "member/infoModify";

}

}

**Member.java**

**import** lombok.AllArgsConstructor;

**import** lombok.Data;

**import** lombok.NoArgsConstructor;

@Data

@AllArgsConstructor

@NoArgsConstructor

**public** **class** Member {

**private** String id;

**private** String pw;

}

**main.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>${greeting }</h1>

<hr>

<h1>board관련 요청 경로</h1>

<h3><a href=*"*${conPath }*/board/write"*>board의 write로 가기</a></h3>

<h3><a href=*"*${conPath }*/board/content"*>board의 content로 가기</a></h3>

<h3><a href=*"*${conPath }*/board/list"*>board의 list로 가기</a></h3>

<h3><a href=*"*${conPath }*/board/reply"*>board의 reply로 가기</a></h3>

<h2></h2>

<hr>

<h1>member관련요청 경로</h1>

<p><button onclick="location.href='member?method=join'">회원가입</button></p>

<p><button onclick="location.href='member?method=login'">로그인</button></p>

<p><button onclick="location.href='member?method=logout'">로그아웃</button></p>

<p><button onclick="location.href='member?method=infoModify'">정보수정</button></p>

</body>

</html>

**views폴더에 board폴더를 만들어 write.jsp생성**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h2>board의 write.jsp페이지입니다</h2>

<hr>

<h2><span onclick="history.back()">메인페이지로가기</span></h2>

</body>

</html>

**views폴더에 board폴더를 만들어 content.jsp생성**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h2>board의 content.jsp입니다</h2>

<h3>ID : ${id }</h3>

<h3>PW : ${pw }</h3>

<h3>member.id : ${member.id }</h3>

<h3>member.pw : ${member.pw }</h3>

<hr>

<h2><span onclick="history.back()">뒤로가기(메인페이지로)</span></h2>

</body>

</html>

**views폴더에 board폴더를 만들어 list.jsp생성**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h2>board의 list.jsp입니다</h2>

<c:if test=*"*${list.size() **eq** 0 }*"*>

<h3>입력된 member가 없습니다</h3>

</c:if>

<c:if test=*"*${list.size() != 0 }*"*>

<c:forEach var=*"member"* items=*"*${list }*"*>

<h3>${member.id } / ${member.pw }</h3>

</c:forEach>

<h3>이상 : ${list.size() }명</h3>

</c:if>

<hr>

<h2><span onclick="history.back()">메인페이지로 가기</span></h2>

</body>

</html>

**views폴더에 board폴더를 만들어 reply.jsp생성**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h2>board의 reply.jsp입니다</h2>

<h2>${member.id } / ${member.pw }</h2>

<hr>

<h2><span onclick="history.back()">뒤로가기(메인페이지로)</span></h2>

</body>

</html>

**views폴더에 member폴더를 만들어 join.jsp생성**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>join.jsp입니다</h1>

<h2>${kind } 가입페이지입니다</h2>

<hr>

<button onclick="history.go(-1);">뒤로가기</button>

</body>

</html>

**views폴더에 member폴더를 만들어 infoModify.jsp, login.jsp, logou.jsp생성**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h2>정보수정 페이지입니다</h2>

<hr>

<button onclick="history.go(-1);">뒤로가기</button>

</body>

</html>