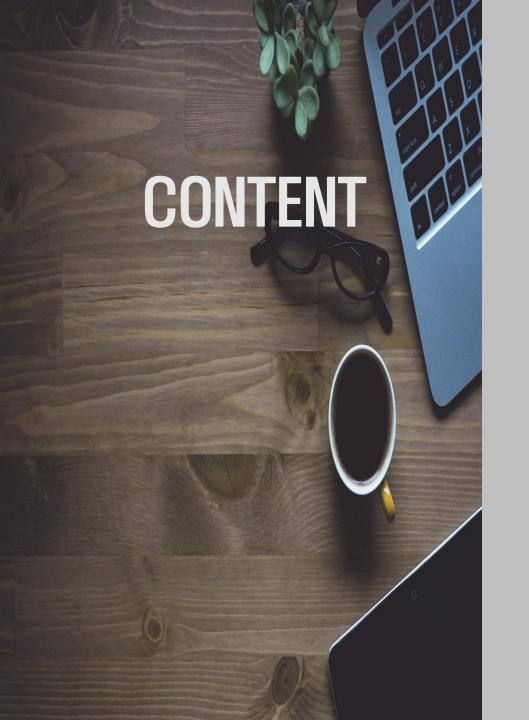
7장 스프링 애너테이션 기능 강사 금영석



스프링 애너테이션이란?

스프링 애너테이션 이용해 URL 요청 실습하기

스프링 애너테이션 이용해 로그 인 기능 구현하기

@Autowired 이용해 빈 주입하기

1. 스프링 애너테이션이란?

- ✓ 스프링 애너테이션(Annotation)
 - 기존에 XML에서 빈 설정을 애너테이션을 이용해서 자바 코드에서 설정하는 방법이다.
 - 기능이 복잡해짐에 따라 XML에서 설정하는 것보다 유지 보수에 유리하다.
 - 현재 애플리케이션 개발시 XML 설정 방법과 애너테이션 방법을 혼합해서 사용한다.

✓ 스프링 애너테이션 제공 클래스

클래스	기능
DefaultAnnotationHandlerMapping	클래스 레벨에서 @RequestMapping을 처리한다.
AnnotationMethodHandlerAdapter	메서드 레벨에서 @RequestMapping을 처리한다.

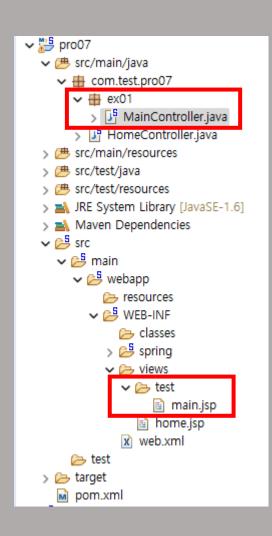


- ✓ <context:component-scan> 태그기능
 - <context:component-scan> 태그를 사용해 패키지 이름을 지정하면 애플리케이션을 실행했을 때 해당 패키지에서 애너테이션으로 지정된 클래스를 빈으로 만들어 준다.

✔ 여러가지 스테레오 애너테이션

애너테이션	기능
@Controller	스프링 컨테이너가 component-scan에 의해 지정한 클래스를 컨트롤러 빈으로 자동 변환한다.
@Service	스프링 컨테이너가 component-scan에 의해 지정한 클래스를 서비스 빈으로 자동 변환한다.
@Repository	스프링 컨테이너가 component-scan에 의해 지정한 클래스를 DAO 빈으로 자동 변환한다.
@Component	스프링 컨테이너가 component-scan에 의해 지정한 클래스를 빈으로 자동 변환한다.

2. 스프링 애너테이션 이용해 URL 요청 실습하기

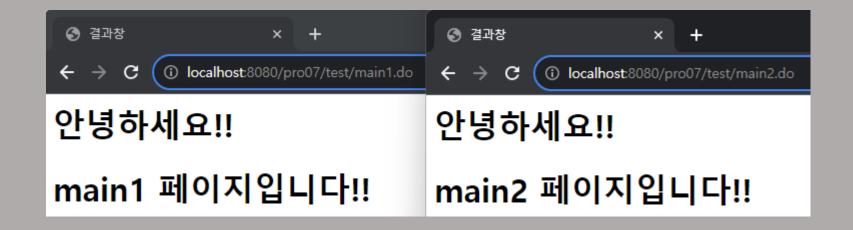


```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"
       isELIgnored="false" %>
    <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5
 6
 7 € < %
     request.setCharacterEncoding("UTF-8");
 9 %>
10
11
12⊖<html>
130<head>
14 <meta charset=UTF-8">
15 <title>결과창</title>
16 </head>
17⊖<body>
18 <h1>안녕하세요!!</h1>
19 <h1>${msq} 페이지입니다!!</h1>
20 </body>
21 </html>
```

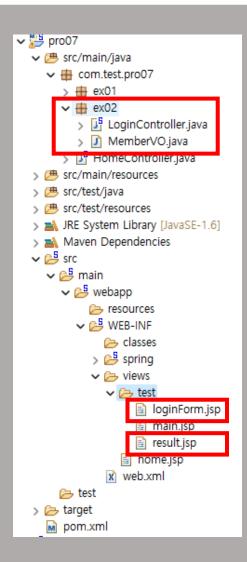


```
1 package com.test.pro07.ex01;
 3@import javax.servlet.http.HttpServletRequest;
 4 import javax.servlet.http.HttpServletResponse;
 6 import org.springframework.stereotype.Controller;
 7 import org.springframework.web.bind.annotation.RequestMapping;
 8 import org.springframework.web.bind.annotation.RequestMethod;
 9 import org.springframework.web.servlet.ModelAndView;
10
11 @Controller("mainController")
12 @RequestMapping("/test")
13 public class MainController {
14⊖
          @RequestMapping (value="/main1.do", method=RequestMethod. GET)
          public ModelAndView main1(HttpServletRequest request, HttpServletResponse response)
15
16
                  throws Exception{
17
             ModelAndView mav=new ModelAndView();
18
             mav.addObject("msg", "main1");
19
             mav.setViewName("main");
20
             return mav;
21
22
23⊜
          @RequestMapping(value="/main2.do", method = RequestMethod. GET)
24
          public ModelAndView main2(HttpServletRequest request, HttpServletResponse response)
25
                  throws Exception{
26
             ModelAndView mav=new ModelAndView();
27
             mav.addObject("msg", "main2");
28
             mav.setViewName("main");
29
             return mav;
30
31 }
```





3. 스프링 애너테이션 이용해 로그인 기능 구현 하기



한글 깨짐 방지를 위한 Filter 적용하기

```
x web.xml ⊠
           <url-pattern>/</url-pattern>
30
       </servlet-mapping>
31
32
33
       <!-- UTF-8 Encoding Filter -->
349
       <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>
           <init-param>
41⊖
42
              <param-name>forceEncoding</param-name>
              <param-value>true</param-value>
43
44
           </init-param>
        </filter>
45
469
        <filter-mapping>
47
           <filter-name>encodingFilter</filter-name>
           <url-pattern>/*</url-pattern>
48
49
        </filter-mapping>
50 </web-app>
```

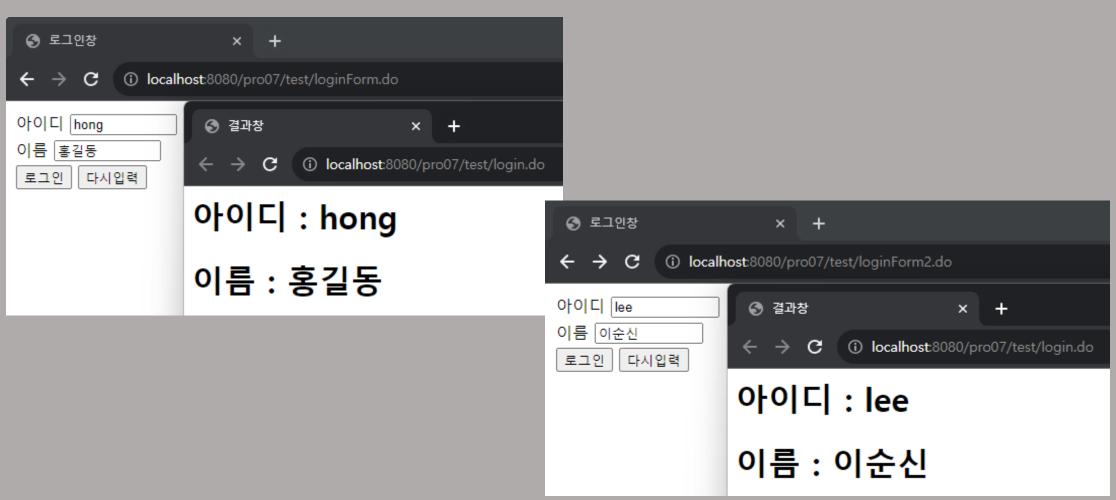
```
1 package com.test.pro07.ex02;
 3 import javax.servlet.http.HttpServletRequest;
10
11 @Controller("loginController")
12 public class LoginController {
13⊜
       @RequestMapping(value = { "/test/loginForm.do", "/test/loginForm2.do" },
14
               method = { RequestMethod.GET })
15
       public ModelAndView loginForm(HttpServletRequest request, HttpServletResponse response)
16
               throws Exception {
17
           ModelAndView mav = new ModelAndView();
18
           mav.setViewName("test/loginForm");
19
           return mav;
20
2.1
22⊖
       @RequestMapping(value = "/test/login.do", method={RequestMethod.GET, RequestMethod.POST})
23
       public ModelAndView login(HttpServletRequest request, HttpServletResponse response)
24
               throws Exception {
25
           request.setCharacterEncoding("utf-8");
26
           ModelAndView mav = new ModelAndView();
27
28
           mav.setViewName("test/result");
           String userID = request.getParameter("userID");
29
           String userName = request.getParameter("userName");
30
           mav.addObject("userID", userID);
31
           mav.addObject("userName", userName);
32
33
           return mav;
34
35 }
```



```
🖹 *loginForm.jsp 🔀
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"
      isELIgnored="false" %>
4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
5 <c:set var="contextPath" value="${pageContext.request.contextPath}"/>
 69<%
    request.setCharacterEncoding("UTF-8");
 8 %>
 9 <!DOCTYPE html>
10⊖<html>
110<head>
12 <meta charset="UTF-8">
13 <title>로그인창</title>
14 </head>
15@<body>
160<form method="post" action="${contextPath}/test/login.do">
      <!-- <input type="hidden" name="email" value="hong@test.com" /> -->
17
      18⊖
19⊜
          20
             Of OF I < input type="text" name="userID" size="10">
21
          229
          23
             >0를<input type="text" name="userName" size="10">
          24
25⊖
          269
             27
               <input type="submit" value="로그연">
28
               <input type="reset" value="다시일력">
29
             30
          31
      32 </form>
33 </body>
34 </html>
```

```
i result.jsp ⋈
 1 < @ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"
       isELIgnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 € < %
     request.setCharacterEncoding("UTF-8");
 7 %>
 80<html>
 90<head>
10 <meta charset=UTF-8">
11 <title>결과창</title>
12 </head>
130<body>
14 <h1>아이디: ${userID }</h1>
15 <h1>이름 : ${userName }</h1>
16@<%-- <h1>아이디: ${info.userID }</h1>
17 <h1>이름 : ${info.userName }</h1> --%>
18 </body>
19 </html>
```





Quiz

- ✓ RequestMapping 으로 아래와 같이 작성 해 보세요
 - localhost:8080/pro07/quiz01/valueSelect.do 와 /quiz01/result.do 를 url 로 연결 해 보세요.
 - valueSelect.do 를 주소창에 입력하면 콜라, 커피, 코코아, 우유 네 가지를 선택 또는 버튼을 클릭하도록 구현 해 보세요.
 - 클릭된 내용은 result.do 로 전송 되어 각 음료의 이름으로 다음과 같이 출력 되도록 해 보세요. '0000 음료 주문이 완료 되었습니다.'



- ✓ @RequestParam으로 Parameter 값 가져오기
 - 매개 변수의 수가 많아지면 일일이 getParameter() 메서드를 적용하기는 많이 불편하다.
 - @RequestParam 메서드를 적용하면 getParameter() 메서드로 받아오던 값을 쉽게 얻을 수 있다.



```
✓ № pro07

→ 

⊕ com.test.pro07

                                       > 🌐 ex01
                                      > # ex02

✓ 

⊕ ex03

                                                    > MemberVO.java
                                                  > 🖟 ParamController.java
                                       > 🖟 HomeController.java
             > # src/main/resources
            > 🌁 src/test/java
             > # src/test/resources
            JRE System Library [JavaSE-1.6]
            > Maven Dependencies
            🗸 🐸 src
                        🗸 🐸 main

✓ № webapp

                                                                 resources

✓ № WEB-INF

                                                                               classes
                                                                  > 🐸 spring
                                                                views
                                                                                                         loginForm.jsp
                                                                                                           main.jsp
                                                                                                           result.jsp
                                                                                              home.jsp
                                                                                 x web.xml
                                      test
            > 📂 target

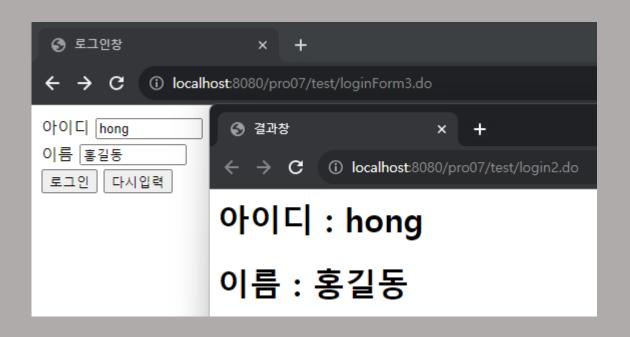
    m pom.xml
    m pom.x
```

```
IoginForm.jsp ⋈
 1 <%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"
      isELIgnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 <c:set var="contextPath" value="${pageContext.request.contextPath}"/>
 69<%
 7 request.setCharacterEncoding("UTF-8");
 8 %>
 9 <!DOCTYPE html>
100<html>
110<head>
   <meta charset="UTF-8">
   <title>로그인창</title>
14 </head>
150<body>
160<form method="post" action="${contextPath}/test/login2.do"
      17⊖
18⊜
          19
             OfO|C| <input type="text" name="userID" size="10">
2.0
          21⊖
          22
             Off <input type="text" name="userName" size="10">
          23
249
          25⊖
             26
               <input type="submit" value="로그인">
27
               <input type="reset" value="다시입력">
28
             29
          30
      31 </form>
32 </body>
33 </html>
```



```
🚺 *ParamController.java 🛭
 1 package com.test.pro07.ex03;
 3 import javax.servlet.http.HttpServletRequest;
11
12 @Controller
13 public class ParamController {
       @RequestMapping(value = "/test/login2.do", method = { RequestMethod.GET, RequestMethod.POST })
14⊖
15
       public ModelAndView login2 (@RequestParam("userID") String userID,
16
                                 @RequestParam("userName") String userName,
17
                                 HttpServletRequest request, HttpServletResponse response)
18
                                         throws Exception {
19
           request.setCharacterEncoding("utf-8");
           ModelAndView mav = new ModelAndView();
20
           mav.setViewName("test/result");
21
22
23
           // String userID = request.getParameter("userID");
24
           // String userName = request.getParameter("userName");
25
26
           System.out.println("userID: "+userID);
           System.out.println("userName: "+userName);
27
28
           mav.addObject("userID", userID);
29
           mav.addObject("userName", userName);
30
31
           return mav;
32
33⊜
       @RequestMapping(value = "/test/loginForm3.do", method = { RequestMethod.GET })
34
       public ModelAndView loginForm(HttpServletRequest request, HttpServletResponse response)
35
               throws Exception {
36
           ModelAndView mav = new ModelAndView();
37
           mav.setViewName("test/loginForm");
38
           return mav:
39
40 }
```







- ✓ @RequestParam의 required 속성 사용하기
 - @RequestParam 을 적용했을 때 required 속성의 기본값은 true 이다.
 - required 속성의 기본값이 true 일 경우 반드시 매개 변수의 값이 존재 해야 한다. 만약 매개변수가 없으면 예외가 발생한다.
 - required 속성을 false로 설정하면 메서드를 호출 했을 때 지정한 이름의 매개변수가 전 달되면 값을 저장하고 없으면 null 값을 저장하고 예외가 발생하지 않는다.



```
→ 

⊕ com.test.pro07

      > # ex01
      > # ex02
     > # ex03
     > D ParamRequiredController.java
     > 🛂 HomeController.java
  > # src/main/resources
  > 🌁 src/test/java
 > # src/test/resources
 > A JRE System Library [JavaSE-1.6]
 > Maven Dependencies
 🗸 🚝 src
   🗸 🐸 main

✓ № webapp

          resources

✓ № WEB-INF

            classes
          > 🐸 spring
          views
            🗸 🗁 test
               loginForm.jsp
               i≣i main.isp
               result.jsp
              nome.jsp
            x web.xml
     test
 > 📂 target
    m pom.xml
```

```
i result.jsp ⋈
 1 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"
       isELIgnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 59<%
     request.setCharacterEncoding("UTF-8");
 7 %>
 80<html>
 9⊖<head>
10 <meta charset=UTF-8">
11 <title>결과창</title>
12 </head>
130<body>
14 <h1>아이디: ${userID }</h1>
15 < h1 > 0 = \cdot $\{userName\} < /h1 >
16 <h1>이메일: ${email }</h1>
17 </body>
18 </html>
```

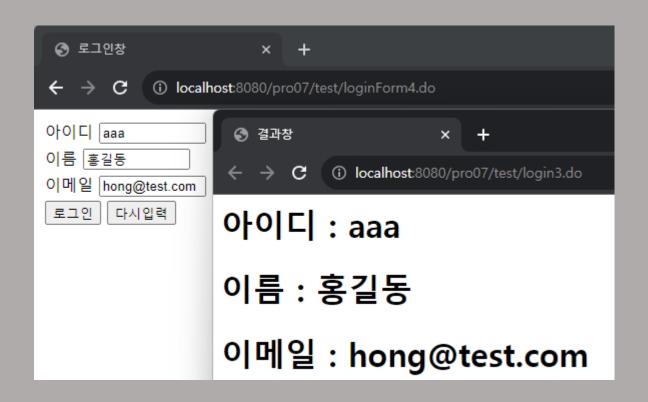


```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
     pageEncoding="UTF-8"
     isELIgnored="false" %>
4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
5 <c:set var="contextPath" value="${pageContext.request.contextPath}"/>
 69<8
7 request.setCharacterEncoding("UTF-8");
8 %>
9 <!DOCTYPE html>
10⊖<html>
110<head>
12 <meta charset="UTF-8">
13 <title>로그인창</title>
14 </head>
15⊖<body>
160<form method="post" action="${contextPath}/test/login3.do">
      17⊖
18⊖
         OPOICI < input type="text" name="userID" size="10">
19
20
         21⊖
         22
23
        24⊖
         25
            Outline type="email" name="email" size="10">
26
         27⊖
         28⊜
29
             <input type="submit" value="로그인">
30
             <input type="reset" value="다시일력">
31
            32
         33
     34 </form>
35 </body>
36 </html>
```



```
1 package com.test.pro07.ex04;
 2
 3 import javax.servlet.http.HttpServletRequest;
11
12 @Controller
13 public class ParamRequiredController {
       @RequestMapping(value = "/test/login3.do", method = { RequestMethod.GET, RequestMethod.POST })
15
       public ModelAndView login (@RequestParam String userID,
16
               @RequestParam(value="userName", required=true) String userName,
17
               @RequestParam(value="email", required=false) String email,
18
               HttpServletRequest request, HttpServletResponse response)
19
               throws Exception {
20
           ModelAndView mav = new ModelAndView();
21
           mav.setViewName("test/result");
22
23
           // String userID = request.getParameter("userID");
24
           // String userName = request.getParameter("userName");
25
26
           System.out.println("userID: "+userID);
           System.out.println("userName: "+userName);
28
           System.out.println("email : " + email);
29
           mav.addObject("userID", userID);
30
           mav.addObject("userName", userName);
           mav.addObject("email", email);
31
32
33
           return mav;
34
35⊜
       @RequestMapping(value = "/test/loginForm4.do", method = { RequestMethod.GET })
36
       public ModelAndView loginForm(HttpServletRequest request, HttpServletResponse response)
37
               throws Exception {
38
           ModelAndView mav = new ModelAndView();
39
           mav.setViewName("test/loginForm");
40
           return mav;
41
42 }
```



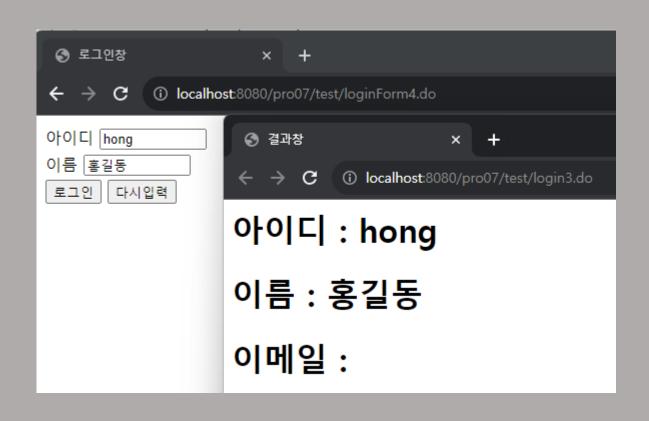




```
    *loginForm.jsp 
    □

pro07/src/main/webapp/WEB-INF/views/test/loginForm.jsp rest page ranguage java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"
      isELIgnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 <c:set var="contextPath" value="${pageContext.request.contextPath}"/>
 6 < %
 7 request.setCharacterEncoding("UTF-8");
 8 %>
 9 <!DOCTYPE html>
10⊖<html>
110<head>
12 <meta charset="UTF-8">
13 <title>로그인창</title>
14 </head>
15⊖<body>
160<form method="post" action="${contextPath}/test/login3.do">
17⊖
       18⊖
           OPOIC <input type="text" name="userID" size="10">
19
20
          21⊖
              Off <input type="text" name="userName" size="10">
22
24
              Outleton type="email" name="email" size="10">
            -->
27⊝
           289
              29
                <input type="submit" value="로그인">
30
                <input type="reset" value="다시일력">
31
              32
          33
       34 </form>
35 </body>
36 </html>
```





Quiz

- ✔ quiz02 패키지를 만들어 아래와 같이 구현 해 보세요.
 - /quiz02/memberJoin.do 로 접근하면 /views/quiz02/memberJoin.jsp 로 이동한다.
 - /views/quiz02/memberJoin.jsp 를 만들고 id, pwd, name, email 을 입력 하고 /quiz02/result.do로 전달 한다.
 - 전달받은 Parameter 값을 /views/quiz02/result.jsp 로 전송 후 출력한다.



- ✓ @RequestParam 을 이용해 Map에 매개 변수 값 설정하기
 - 전송 되는 매개변수의 수가 많을 경우에는 Map에 바로 저장해서 사용하면 편리하다.

```
→ 

⊕ com.test.pro07

   > # ex01
   > # ex02
   > # ex03
   > # ex04
   MapController.java
   > 🛂 HomeController.java
  test
       loginForm.jsp
        main.jsp
        result.isp
       result1.jsp
      home.jsp
    x web.xml
```

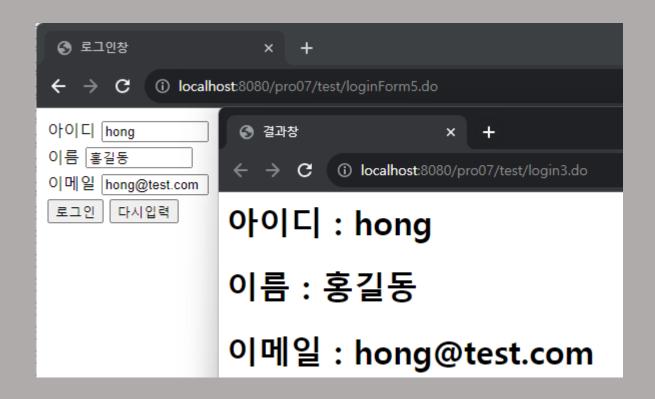
```
📓 result1.jsp 🖂
 1 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"
       isELIqnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 € < %
     request.setCharacterEncoding("UTF-8");
 7 %>
 80<html>
 90<head>
10 <meta charset=UTF-8">
11 <title>결과창</title>
12 </head>
130<body>
14 <h1>아이디: ${info.userID }</h1>
15 <h1>이름 : ${info.userName }</h1>
16 <h1>이메일: ${info.email }</h1>
17 </body>
18 </html>
```



```
IoginForm.jsp ⋈
 1 <%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"
      isELIgnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 <c:set var="contextPath" value="${pageContext.request.contextPath}"/>
 69<8
 7 request.setCharacterEncoding("UTF-8");
 8 %>
 9 <!DOCTYPE html>
10⊖<html>
110<head>
12 <meta charset="UTF-8">
13 <title>로그인창</title>
14 </head>
150<body>
160<form method="post" action="${contextPath}/test/login4.do"
      17⊖
18⊖
          19
             OfOICI <input type="text" name="userID" size="10">
20
          21⊖
          22
             0 = 'input type="text" name="userName" size="10">
23
24⊖
          25
             Outline type="email" name="email" size="10">
         26
27⊖
          28⊜
29
               <input type="submit" value="]</pre>
               <input type="reset" value="다시일력">
30
31
             32
          33
      34 </form>
35 </body>
36 </html>
```

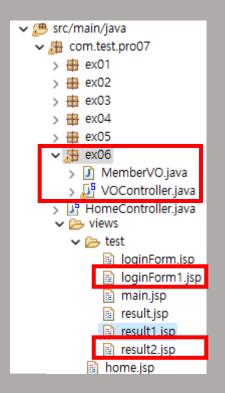
```
1 package com.test.pro07.ex05;
 3 import java.util.Map;
13
14 @Controller
15 public class MapController {
16⊖
       @RequestMapping(value = "/test/login4.do", method = { RequestMethod. GET, RequestMethod.POST })
17
       public ModelAndView login(@RequestParam Map<String, String> info,
18
               HttpServletRequest request, HttpServletResponse response)
19
               throws Exception {
20
           ModelAndView mav = new ModelAndView();
21
           mav.setViewName("test/result1");
22
23
           System.out.println("userID: "+ info.get("userID"));
24
           System.out.println("userName: "+info.get("userName"));
25
           System.out.println("email : " + info.get("email"));
26
27
           mav.addObject("info", info);
28
29
           return mav;
30
31⊜
       @RequestMapping(value = "/test/loginForm5.do", method = { RequestMethod.GET })
32
       public ModelAndView loginForm(HttpServletRequest request, HttpServletResponse response)
33
               throws Exception {
34
           ModelAndView mav = new ModelAndView();
35
           mav.setViewName("test/loginForm");
36
           return mav;
37
38 }
```







- ✓ @ModelAttribute 이용해 VO에 매개변수 값 설정하기
 - Map 은 여러 개의 값을 저장하지만 @ModelAttribute 를 이용하면 VO 객체의 멤버 변수 와 일치하는 Parameter 값을 자동으로 저장하여 편리하다.



```
i result2.jsp ⋈
 1 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"
       isELIqnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 € < %
 6 request.setCharacterEncoding("UTF-8");
 7 %>
 8⊖<html>
 90<head>
10 <meta charset=UTF-8">
11 <title>결과창</title>
12 </head>
130 < body>
14 <h1>아이디: ${member.id }</h1>
15 <h1>암호 : ${member.pwd }</h1>
16 <h1>이름 : ${member.name }</h1>
17 <h1>이메일: ${member.email }</h1>
18 </body>
19 </html>
```

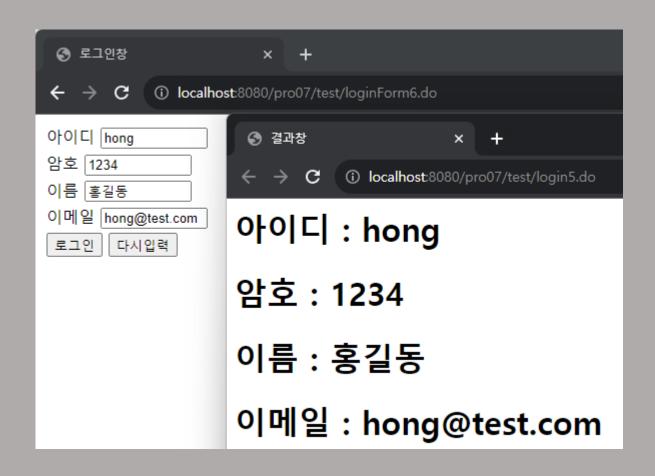


```
IoginForm1.jsp ⋈
 1 <%@ page language="java" contentType="text/html; charset=UTF-8"
 pageEncoding="UTF-8"
     isELIgnored="false" %>
 4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5 <c:set var="contextPath" value="${pageContext.request.contextPath}"/>
 7 request.setCharacterEncoding("UTF-8");
 8 %>
 9 <!DOCTYPE html>
10⊖<html>
11⊖<head>
12 <meta charset="UTF-8">
13 <title>로그인창</title>
14 </head>
15⊖<body>
160 < form method="post" action="${contextPath}/test/login5.do">
      18⊖
         19
            Of O C I content type="text" name="id" size="10">
20
         21⊖
         22
            악호 <input type="passwd" name="pwd" size="10">
23
         24⊖
         25
            26
         27⊝
         28
            olub <input type="email" name="email" size="10">
29
         30⊝
         31⊖
            32
              <input type="submit" value="로그인">
33
              <input type="reset" value="다시일력">
34
            35
         36
      37 </form>
38 </body>
39 </html>
```



```
🕡 *VOController.java 🖂
   package com.test.pro07.ex06;
3. 3. import javax.servlet.http.HttpServletRequest;
13 @Controller
14 public class VOController {
15⊜
       @RequestMapping(value = "/test/login5.do", method = { RequestMethod.GET, RequestMethod.POST })
16
       public ModelAndView login (@ModelAttribute("vo") MemberVO member,
17
               HttpServletRequest request, HttpServletResponse response)
18
               throws Exception {
19
           ModelAndView mav = new ModelAndView();
20
           mav.setViewName("test/result2");
21
22
           System.out.println("아이디: " + member.getId());
23
           System.out.println("암호: " + member.getPwd());
24
           System.out.println("이름: " + member.getName());
25
           System.out.println("이메일: " + member.getEmail());
26
27
           mav.addObject("member", member);
28
29
            return mav;
30
31⊖
       @RequestMapping(value = "/test/loginForm6.do", method = { RequestMethod.GET })
32
       public ModelAndView loginForm(HttpServletRequest request, HttpServletResponse response)
33
               throws Exception {
34
           ModelAndView mav = new ModelAndView();
35
           mav.setViewName("test/loginForm1");
36
           return mav;
37
38 }
```







- ✔ Model 클래스를 이용해 값 전달 하기
 - Model 클래스를 이용하면 메서드를 호출 했을 때 JSP로 값을 바로 바인딩하여 전달할 수 있다.
 - Model 클래스의 addAttribute() 메서드는 ModelAndView의 addObject() 메서드와 같은 기능을 수행 한다.
 - Model 클래스는 따로 View 정보를 전달할 필요가 없을 때 사용하면 편리하다.



```
√ I com.test.pro07

    > # ex01
    > # ex02
    > # ex03
    > # ex04
    > # ex05
    > Æ ex06

✓ Æ ex07

       > 15 ModelController.java
     > 🛂 HomeController.java
     views

✓ 

best

test

            loginForm.jsp
            loginForm1.jsp
            result.jsp
            □ result i.jsp
            result2.jsp
          home.jsp
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
       pageEncoding="UTF-8"
       isELIqnored="false" %>
    <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
 5⊖<%
     request.setCharacterEncoding("UTF-8");
 7 응>
 8⊖<html>
 9⊖<head>
10 <meta charset=UTF-8">
11 <title>결과창</title>
12 </head>
13⊖<body>
14 <h1>아이디: ${userID }</h1>
15 <h1>이름 : ${userName }</h1>
16 <%-- <h1>이메일: ${email }</h1> --%>
17 </body>
18 </html>
```



```
🔝 *ModelController.java 🔀
 1 package com.test.pro07.ex07;
3. import javax.servlet.http.HttpServletRequest;
13
14 @Controller
15 public class ModelController {
16⊜
        @RequestMapping(value = "/test/login6.do", method = { RequestMethod.GET, RequestMethod.POST })
        public String login (Model model,
18
                HttpServletRequest request, HttpServletResponse response)
                throws Exception {
19
20
21
22
23
            model.addAttribute("userID", "hong");
                                                                      결과창
            model.addAttribute("userName", "홍길동");
                                                                               ① localhost:8080/pro07/test/login6.do
24
25
            return "test/result";
                                                                     아이디: hong
26 }
                                                                     이름: 홍길동
```

4. @Autowired 이용해 빈 주입하기>

- ✓ @Autowired 의 특징
 - 기존 XML 파일에서 각각의 빈을 DI로 주입 했던 기능을 코드에서 애너테이션으로 자동으로 수행한다.
 - @Autowired를 사용하면 별도의 setter나 생성자 없이 속성에 빈을 주입할 수 있다는 특징이 있다.



```
      Y # src/main/java

      V # com.test.pro07

      ⇒ ex01

      ⇒ ex02

      ⇒ ex03

      ⇒ ex04

      ⇒ ex05

      ⇒ ex06

      ⇒ ex07

      V # ex08

      ⇒ # AutoController.java

      > # AutoService.java

      > # MemberVO.java

      > # HomeController.java
```

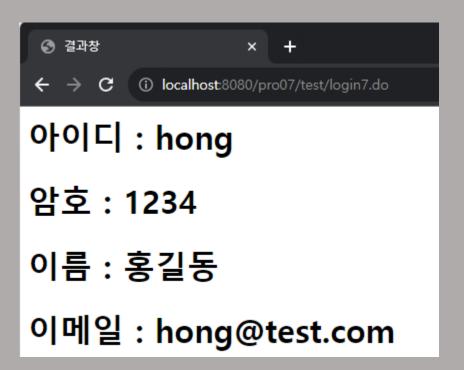
```
1 package com.test.pro07.ex08;
 3⊕ import javax.servlet.http.HttpServletRequest;
12 @Controller
13 public class AutoController {
       @Autowired
14⊜
15
       AutoService service;
16
17⊝
       @RequestMapping(value = "/test/login7.do", method = { RequestMethod.GET, RequestMethod.POST })
18
       public ModelAndView login(HttpServletRequest request, HttpServletResponse response)
19
               throws Exception {
20
           ModelAndView mav = new ModelAndView();
21
           mav.setViewName("test/result2");
22
23
           MemberVO member = service.selectMember();
24
25
           mav.addObject("member", member);
26
27
           return mav;
28
29 }
```



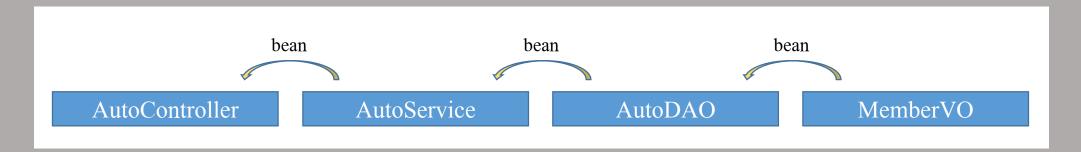
```
1 package com.test.pro07.ex08;
 3 import org.springframework.beans.factory.annotation.Autowired;
  5
  6 @Service
 7 public class AutoService {
       @Autowired
 8⊜
                                                  9
       AutoDAO dao;
                                                     package com.test.pro07.ex08;
10
                                                    2
11⊖
       public MemberVO selectMember() {
                                                    3 import org.springframework.beans.factory.annotation.Autowired;
212
           // TODO Auto-generated method stub
           MemberVO member = dao.selectMember();
13
                                                    6 @Repository
14
           return member;
                                                    7 public class AutoDAO {
15
                                                          @Autowired
16
                                                          MemberVO member;
17 }
                                                   10
                                                  11⊖
                                                          public MemberVO selectMember() {
                                                 212
                                                             // TODO Auto-generated method stub
                                                             member.setId("hong");
                                                  13
                                                  14
                                                             member.setPwd("1234");
                                                  15
                                                             member.setName("홍길동");
                                                  16
                                                             member.setEmail("hong@test.com");
                                                  17
                                                             return member;
                                                  18
                                                  19 }
```



```
1 package com.test.pro07.ex08;
 2
 3⊕ import java.sql.Date;
   @Component
   public class MemberV0 {
       private String id;
       private String pwd;
10
11
       private String name;
12
       private String email;
13
       private Date joinDate;
14⊖
       public String getId() {
15
           return id;
16
17⊝
       public void setId(String id) {
           this.id = id;
18
19
```



✔ @Autowired로 주입한 빈 상태는 아래와 같다.



Quiz

✓ pro06 프로젝트의 member 이하의 모든 패키지를 가져와서 @Autowired 로 변경 해 보세요.

