

Servlet & JSP Exercise 회원 관리

DB에 저장되어 있는 데이터를 읽어 회원의 로그인 정보를 확인하고, 전체 회원 정보 조회 및 로그아웃 기능을 제공하는 간단한 시스템을 작성 한다.

목적

- Deployment Descriptor
- HttpSession
- JSP Pages Tags
- Connection Pool
- DAO Pattern
- Model 2 Architecture

1. 개요

"exercise" 컨텍스트로 Model 2 Architecture 웹 어플리케이션을 개발한다.

DBMS로 정보를 관리하며, 컨텍스트가 시작될 때 컨텍스트 리스너 클래스가 DB Connection Pool을 관리하는 DBService 시스템을 작동시킨다.

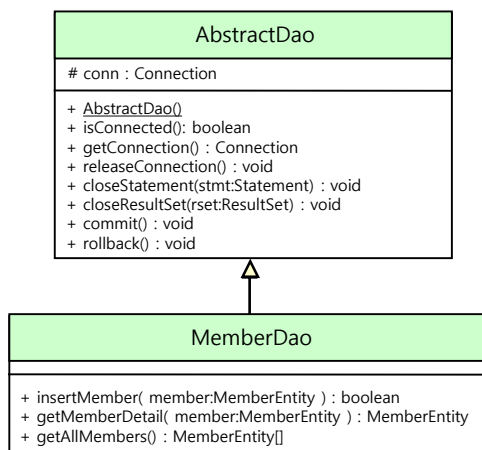
멤버에 대한 JDBC 작업을 담당하는 DAO 클래스를 별도로 작성하고, Biz 클래스에서는 DAO 클래스가 제공하는 기능을 이용하여 멤버 등록, 로그인, 전체 멤버조회, 로그아웃 등의 비즈니스 서비스 작업을 수행한다. 각각의 서블릿은 Biz 클래스를 이용하는 controller의 역할을 담당하며, View 역할을 담당하는 JSP 페이지도 작성하도록 한다.

2. Class Diagram

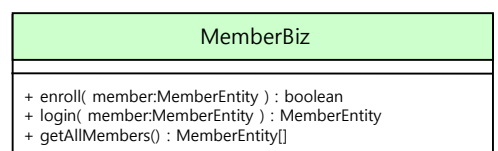
< Entity Class >



< DAO Class >

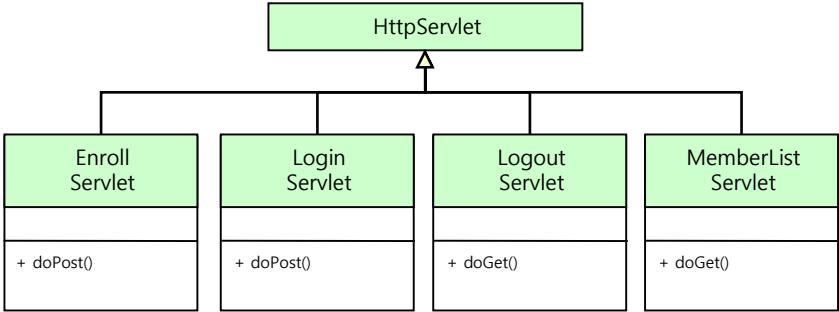


< Biz Class >



2. Class
Diagram(계속)

< Controller Class >



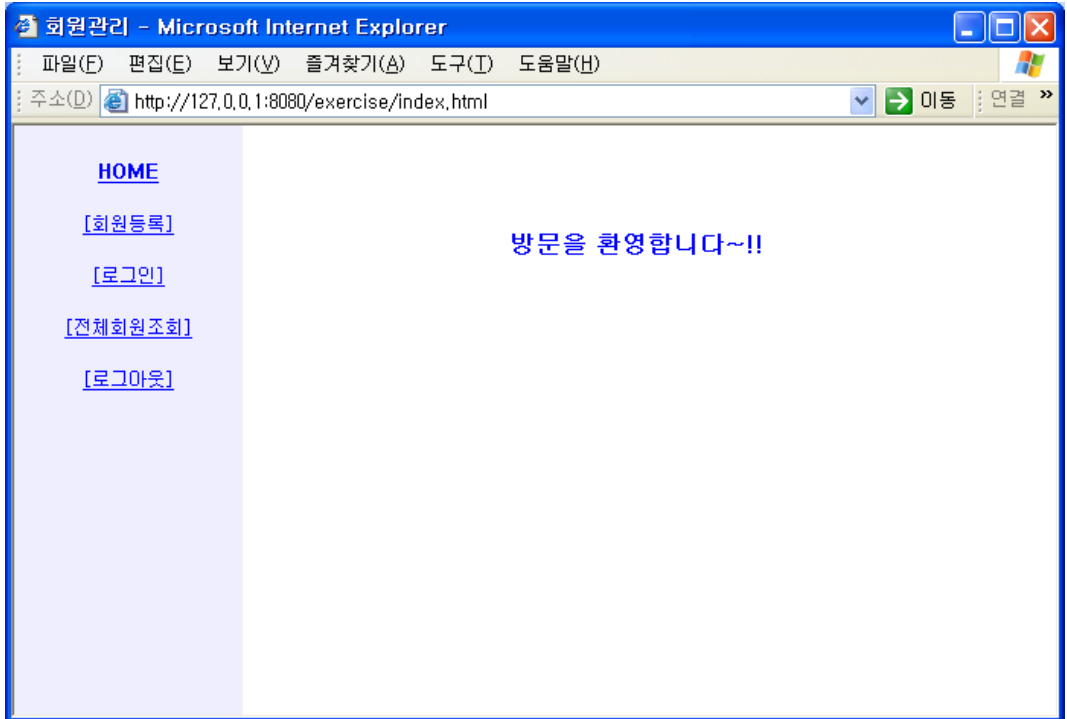
3. DB Table

❖ CS_MEMBER : 회원 정보를 저장하는 테이블

Column	Data type	Null 여부	Column 설명
MEMBER_ID	VARCHAR2(12)	NOT NULL,PK	멤버 아이디
NAME	VARCHAR2(40)	NOT NULL	멤버 이름
PASSWORD	VARCHAR2(12)	NOT NULL	로그인 암호
ADDRESS	VARCHAR2(100)		주소
PHONE	VARCHAR2(15)		전화번호
PASSWORD_QUESTION	VARCHAR2(100)		본인확인 힌트
PASSWORD_ANSWER	VARCHAR2(100)		본인확인 답변
MARRIAGE	CHAR(1)		결혼여부
HOBBY	VARCHAR2(40)		취미
ETC	VARCHAR2(100)		비고

4. Sample Run

① <http://본인IP:port/exercise> 로 접속하면 보이는 초기 화면이다.



② [회원등록]을 선택했을 때 보이는 회원가입 화면이다.

4. Sample Run
(계속)

③ 회원 정보를 입력하고 작성완료 버튼을 선택한다.

회원관리 - Microsoft Internet Explorer

주소(D) http://127.0.0.1:8080/exercise/index.html

HOME

[회원등록]

[로그인]

[전체회원조회]

[로그아웃]

@@ 회원 가입 @@

ID	twin4u
이름	홍길동
암호
주소	서울시 마포구 상암동 B5-3
전화번호	02-6710-1234
본인확인 힌트	배우자의 이름은?
본인확인 답변	김태희
결혼 여부	<input checked="" type="radio"/> 기혼 <input type="radio"/> 미혼
취미	<input type="checkbox"/> 컴퓨터 <input checked="" type="checkbox"/> 문화 <input checked="" type="checkbox"/> 스포츠 <input type="checkbox"/> 기타
비고	화이팅~

작성완료 다시쓰기

Copyright ChatStar Corp. All rights reserved.

④ 회원등록이 되었다는 메시지와 함께 등록한 회원 정보가 나타난다.

회원관리 - Microsoft Internet Explorer

주소(D) http://127.0.0.1:8080/exercise/index.html

HOME

[회원등록]

[로그인]

[전체회원조회]

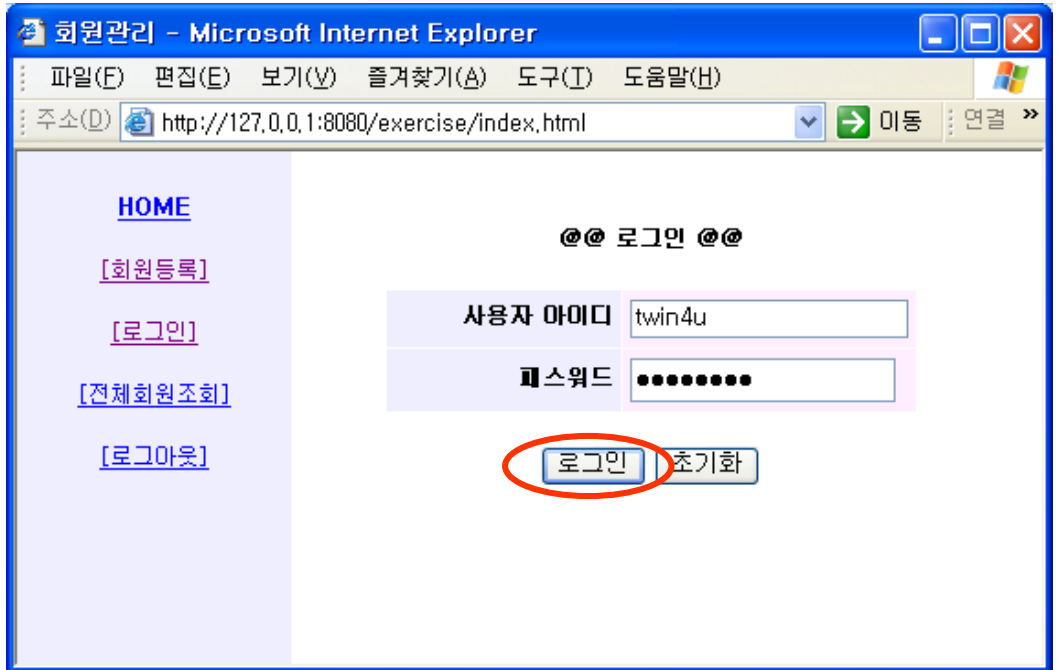
[로그아웃]

아래와 같이 회원이 등록되었습니다!!

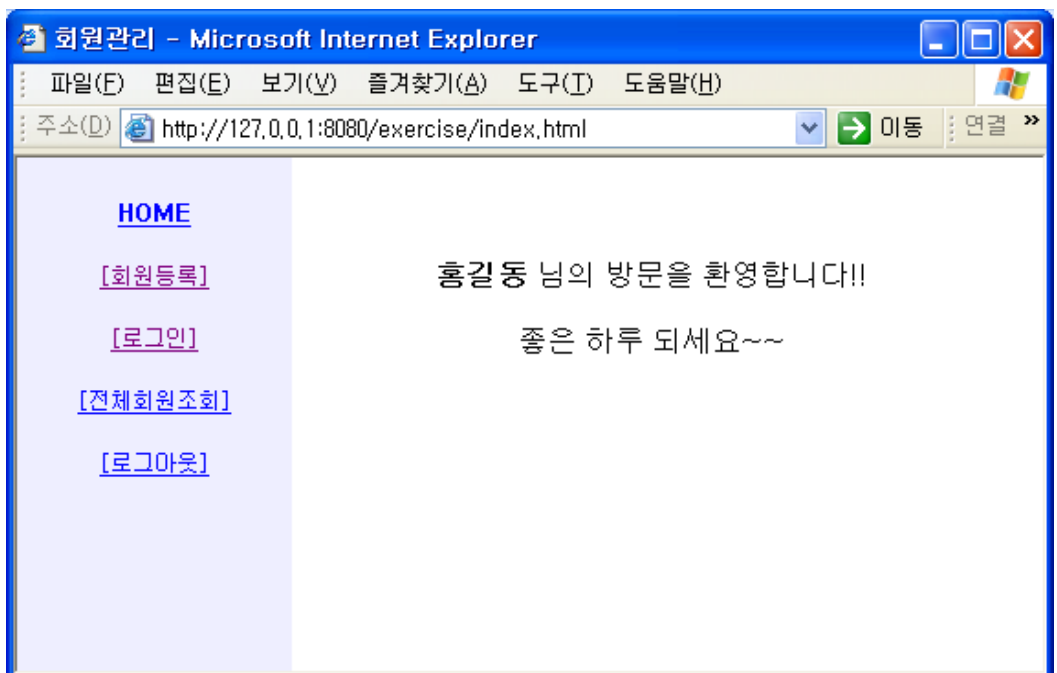
ID	twin4u
이름	홍길동
암호	test1234
주소	서울시 마포구 상암동 B5-3
전화번호	02-6710-1234
본인확인 힌트	배우자의 이름은?
본인확인 답변	김태희
결혼 여부	Y
취미	문화 스포츠
비고	화이팅~

4. Sample Run
(계속)

⑤ [로그인] 을 선택하고 등록되어 있는 회원 아이디와 패스워드를 입력하여 로그인한다.

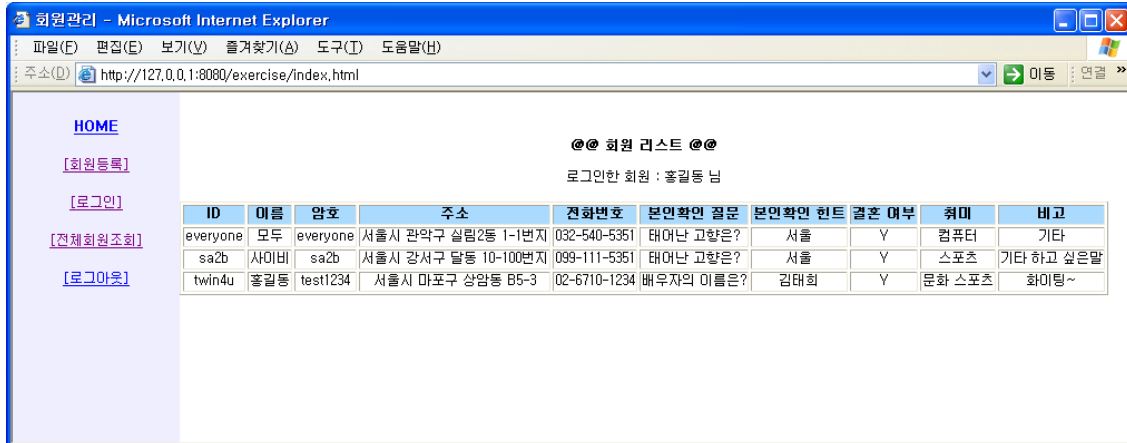


⑥ 로그인 아이디와 패스워드가 일치하면 회원 이름과 함께 환영 메시지가 나타난다.

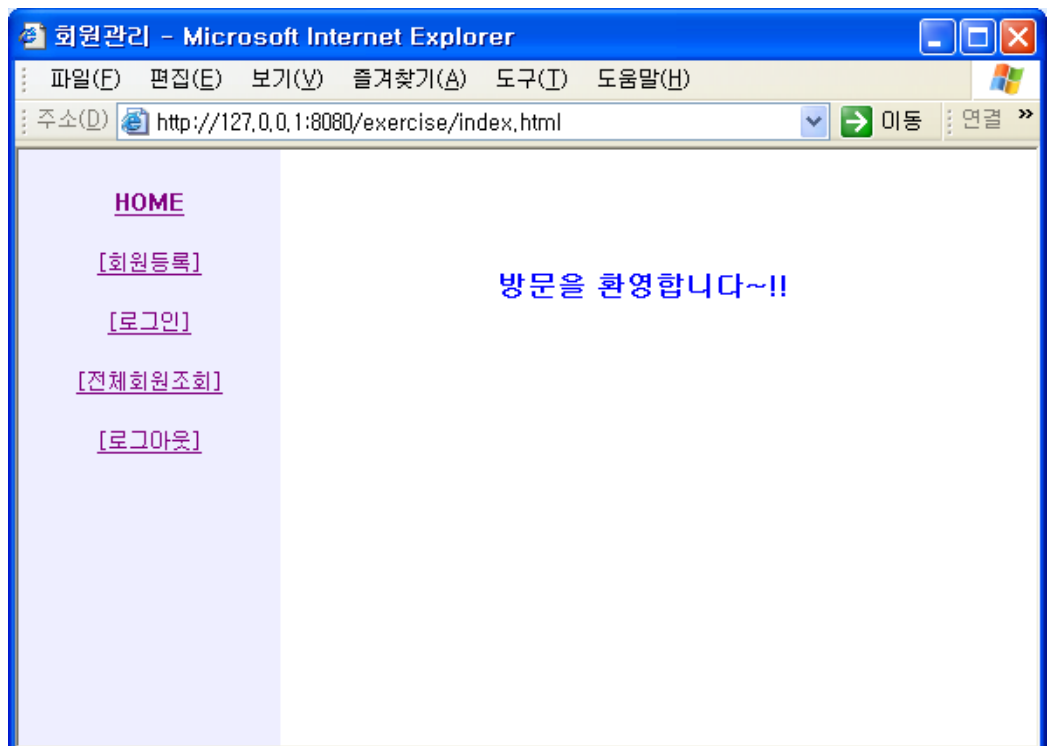


4. Sample Run
(계속)

- ⑦ 로그인 후, [전체회원조회]를 선택하면 로그인한 회원의 이름과 전체 회원 리스트가 화면에 나타난다.

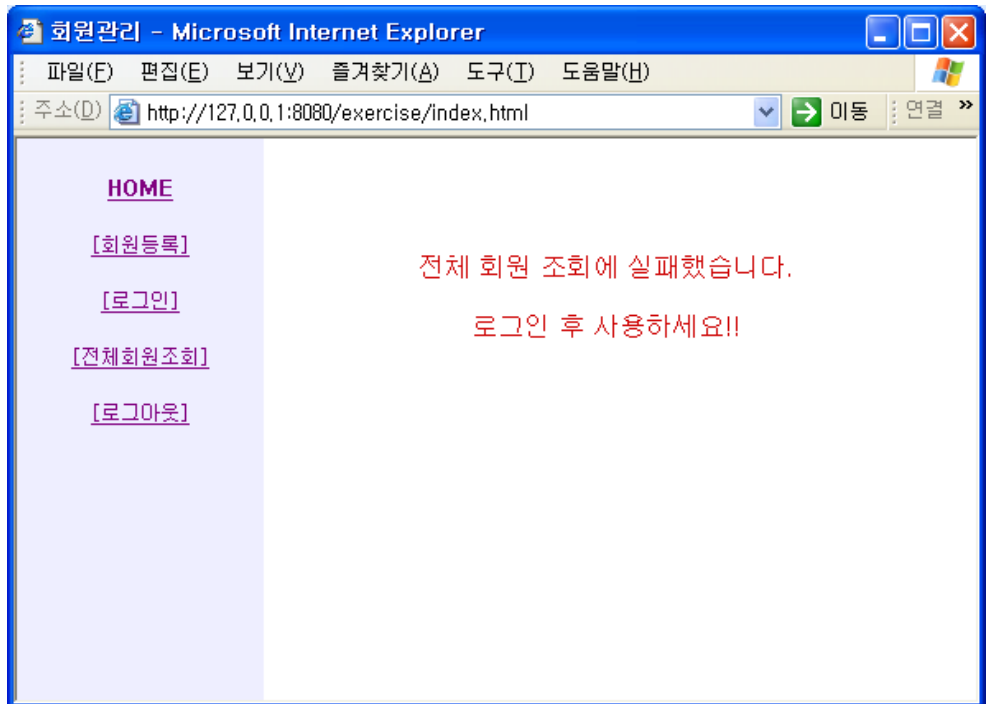


- ⑧ [로그아웃]을 선택하면 로그아웃이 되면서 세션에 담겨진 회원정보가 사라지고 초기 화면이 나타난다.

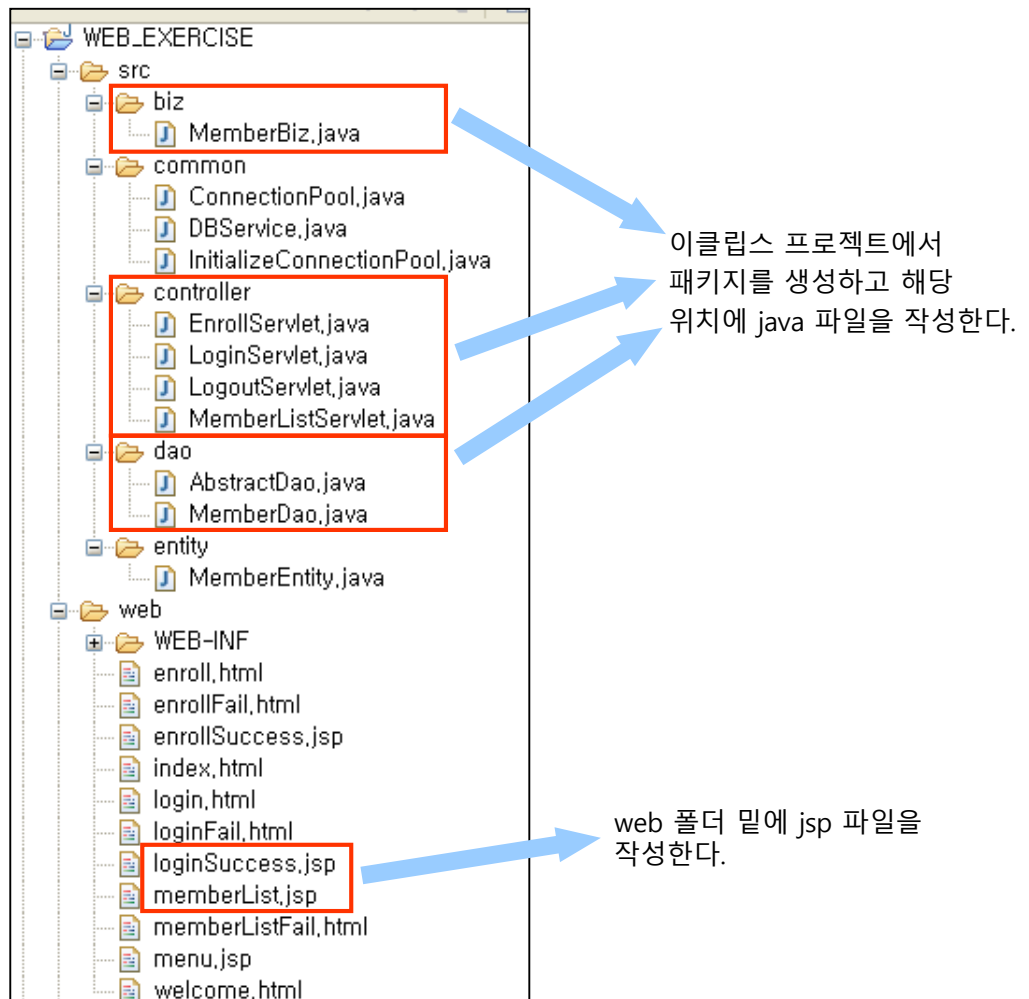


4. Sample Run (계속)

- ㉠ 로그아웃을 한 후, [전체회원조회]를 다시 선택하면 전체 회원조회가 되지 않고 로그인 후에 사용하라는 메시지가 나타난다.



5. 작성할 파일



6. 코딩 가이드

❖ EnrollServlet.java (부분제공)

: 회원 등록을 처리하는 controller 서블릿 클래스

1. 회원 가입이 성공했을 때 이동할 파일과 실패했을 때 이동할 파일을 작성한다.

❖ LoginServlet.java

: 로그인을 처리하는 controller 서블릿 클래스

1. package는 controller로 설정하고, 필요한 클래스들을 import 한다.
2. doPost() : 클라이언트의 POST 방식 요청을 처리하는 메소드
 - 1) request의 한글처리를 위해 setCharacterEncoding 메소드를 사용한다.
 - 2) 클라이언트로부터 입력 받은 로그인 정보를 저장하기 위해 MemberEntity 객체 member를 생성한다.
 - 3) FORM 데이터로 읽어온 loginId와 password의 값을 위에서 생성한 MemberEntity에 저장한다.
 - 4) MemberBiz 객체 memberBiz를 생성한다.
 - 5) 사용자의 로그인 정보를 가지고 있는 MemberEntity 객체를 전달하면서 MemberBiz 클래스의 로그인 메소드를 호출하고, 결과를 리턴받는다.
 - 6) 리턴 받은 멤버 객체가 null이 아니면(로그인 정보가 valid한 경우), session 영역을 생성하여 "loginMember" 이름으로 MemberEntity 객체를 저장하고, 로그인 성공 화면인 loginSuccess.jsp 로 request를 forward 한다.
(※ RequestDispatcher 객체의 forward 메소드 사용)
 - 7) 리턴 받은 멤버 객체가 null이면(로그인 정보가 invalid한 경우), 로그인 실패 화면인 loginFail.html을 보여준다.
(※ response.sendRedirect("loginFail.html");)

6. 코딩 가이드 (계속)

❖ LogoutServlet.java

: 로그아웃을 처리하는 controller 서블릿 클래스

1. package는 controller로 설정하고, 필요한 클래스들을 import 한다.
2. doGet() : 클라이언트의 GET 방식 요청을 처리하는 메소드
 - 1) 현재 사용하고 있는 session을 얻어온다.
 - 2) session이 존재하면, session을 invalidate 한다.
 - 3) 초기화면인 index.html을 보여준다.

❖ MemberListServlet.java

: 전체 회원 정보를 보여주는 controller 서블릿 클래스

1. package는 controller로 설정하고, 필요한 클래스들을 import 한다.
2. doGet() : 클라이언트의 GET 방식 요청을 처리하는 메소드
 - 1) request의 한글처리를 위해 setCharacterEncoding 메소드를 사용한다.
 - 2) 현재 사용하고 있는 session 정보를 얻어온다.
 - 3) session이 존재하면, session에서 로그인한 회원의 MemberEntity 객체를 가져온다.
 - 4) response의 한글처리를 위해 setContentType 메소드를 사용한다.
 - 5) MemberBiz 객체 memberBiz를 생성한다.
 - 6) 전체 회원을 조회하는 MemberBiz 클래스의 메소드를 호출하고, 결과를 MemberEntity 객체의 배열로 리턴받는다.
 - 7) 리턴받은 MemberEntity 객체의 배열이 null이 아니고 로그인한 회원이 null이 아니면, request 영역을 생성하여 "members" 이름으로 MemberEntity 객체 배열을 저장하고, 전체 회원조회 화면인 memberList.jsp 로 request를 forward 한다.
(※ RequestDispatcher 객체의 forward 메소드 사용)
 - 8) 리턴받은 MemberEntity 객체의 배열이 null이거나 로그인한 회원이 null이면, 회원 조회 실패 화면인 memberListFail.html 을 보여준다.
(※ response.sendRedirect("memberListFail.html ");)

6. 코딩 가이드 (계속)

❖ MemberBiz.java

: 비즈니스 서비스 기능을 담당하는 Model 클래스.

회원 등록, 로그인, 로그아웃, 전체 회원정보 조회 기능을 제공한다.

1. package는 biz로 설정하고, 필요한 클래스들을 import 한다.

2. enroll(MemberEntity member)

1) MemberDao 객체 memberDao를 생성한다.

2) 파라미터로 전달받은 MemberEntity 객체를 전달하면서 DB에 회원을 추가하기 위한 MemberDao 클래스의 메소드를 호출한다.

3) 회원 추가 작업이 성공한 경우에는 commit()을, 실패한 경우에는 rollback()을 수행한다.

4) connection을 반환하고, 등록 성공 여부를 반환한다.

3. login(MemberEntity member)

1) MemberDao 객체 memberDao를 생성한다.

2) 파라미터로 전달받은 MemberEntity 객체를 전달하면서 해당 회원을 조회하는 MemberDao 클래스의 메소드를 호출한다.

3) 조회한 결과로 리턴받은 MemberEntity 객체가 null은 아닌데, 로그인 시 입력한 비밀번호가 DB에 저장되어 있는 비밀번호와 맞지 않으면 해당 MemberEntity 객체를 null로 세팅한다.

4) connection을 반환하고, MemberEntity 객체의 최종 결과를 리턴한다.

4. MemberEntity[] getAllMembers()

1) MemberDao 객체 memberDao를 생성한다.

2) 전체 회원을 조회하는 MemberDao 클래스의 메소드를 호출한다.

3) connection을 반환하고, MemberEntity 객체의 배열을 리턴한다.

6. 코딩 가이드 (계속)

❖ MemberDao.java

: 멤버 정보 DB 작업을 담당하는 DAO 클래스

1. package는 dao로 설정하고, 필요한 클래스들을 import 한다.
2. 제공되는 AbstractDao 클래스를 상속받는 클래스로 선언한다.
3. insertMember(MemberEntity member) : DB에 회원을 추가하는 메소드
 - 1) 파라미터로 전달받은 회원 정보를 DB에 추가한다.
 - 2) 등록 성공 여부를 리턴한다.
4. getMemberDetail(MemberEntity member) : DB에서 특정 회원정보를 조회하는 메소드
 - 1) 파라미터로 전달받은 회원 정보의 회원 ID를 이용해서, DB로 부터 회원 정보를 조회한다.
 - 2) 조회된 회원 정보가 있으면 해당 회원정보를 MemberEntity 객체로 생성하여 리턴하고, 없으면 null을 리턴한다.
5. getAllMembers() : DB에서 전체 회원정보를 조회하는 메소드
 - 1) DB에서 전체 멤버의 정보를 조회한다.
 - 2) 전체 회원 정보가 담긴 MemberEntity 배열을 리턴한다.

❖ loginSuccess.jsp

: 로그인 성공 시 보여주는 View JSP 페이지

1. page의 content type을 text/html로, character set을 euc-kr로 설정한다.
2. 필요한 클래스를 import 하는 page 설정을 추가한다.
3. JSP useBean 태그를 이용하여, session 영역에서 "loginMember"의 이름의 MemberEntity 객체를 얻어온다.
4. 얻어 온 MemberEntity 객체로 부터 로그인한 회원의 이름을 추출하여, Sample Run과 동일한 양식으로 화면에 display 한다.

6. 코딩 가이드 (계속)

❖ memberList.jsp

: 전체 회원 정보 조회 결과를 보여주는 View JSP 페이지

1. page의 content type을 text/html로, character set을 euc-kr로 설정한다.
2. 필요한 클래스를 import 하는 page 설정을 추가한다.
3. 전체 회원 정보를 위해, request 영역에서 "members"라는 이름의 MemberEntity 객체 배열을 가져온다.
4. 현재 로그인한 회원의 정보를 위해, session 영역에서 "loginMember"라는 이름의 MemberEntity 객체를 얻어온다.
5. 로그인한 회원의 정보를 화면에 display 한다. (Sample Run 참조)
6. MemberEntity 객체 배열의 내용을 바탕으로 화면에 보여줄 회원 리스트를, HTML의 table 태그를 사용하여 회원 수 만큼 동적으로 작성한다. (Sample Run 참조)

7. 제공 파일

❖ enroll.sql

: 본 Exercise에서 사용할 CS_MEMBER 테이블과 데이터를 생성하는 SQL 파일

❖ WEB_EXERCISE 프로젝트 : Exercise 작성용 프로젝트

1. Connection Pooling 관련 제공파일
 - src/common/DBService.java
 - src/common/ConnectionPool.java
 - src/common/InitializeConnectionPool.java
2. Entity 클래스
 - src/entity/MemberEntity.java
3. Controller 클래스
 - src/controller/EnrollServlet.java (부분제공)
4. HTML 및 JSP 파일
 - web/index.html : 시스템 초기 화면
 - web/menu.jsp : 화면의 좌측 메뉴를 구성하는 화면
 - web/welcome.html : [HOME] 메뉴를 선택했을 때 나타나는 화면
 - web/enroll.html : [회원등록] 메뉴를 선택했을 때 나타나는 화면
 - web/login.html : [로그인] 메뉴를 선택했을 때 나타나는 화면
 - web/enrollFail.html : 회원 등록이 실패했을 때 나타나는 화면
 - web/enrollSuccess.jsp : 회원 등록이 성공했을 때 나타나는 화면
 - web/loginFail.html : 로그인이 실패했을 때 나타나는 화면
 - web/memberListFail.html : 전체회원 조회가 실패했을 때 나타나는 화면