

11장. 게시판 1

예제1. 게시판의 로그인 전/후 화면 분리하기

Dynamic Web Project : 11-board

[클라이언트]

Folder	: main
JSP File	: WebContent/main/index.jsp
Folder	: member
JSP File	: WebContent/member/loginOk.jsp WebContent/member/loginFail.jsp
Folder	: image
image File	: WebContent/image/tumi.png
Folder	: sql
sql File	: WebContent/sql/member.sql

- 10-member에서 작업한 파일을 그대로 사용한다. => 복사해서 붙여 놓는다.

Folder	: member
JSP File	: WebContent/member/loginForm.jsp (수정) WebContent/member/login.jsp (수정) WebContent/member/writeForm.jsp (수정) WebContent/member/write.jsp WebContent/member/checkId.jsp
Folder	: script
Javascript File	: WebContent/script/memberScript.js

[서버]

- 10-member에서 작업한 파일을 그대로 사용한다. => 복사해서 붙여 놓는다.

Package	: member.bean
Class File	: src/member.bean/MemberDTO.java (Data Transfer Object)
Package	: member.dao
Class File	: src/member.dao/MemberDAO.java (Data Access Object)

<실행 결과>

1. 메인 화면 : 로그인 전

- index.jsp




2. 로그인 확인

- loginForm.jsp, login.jsp, loginOk.jsp, loginFail.jsp, memberScript.js

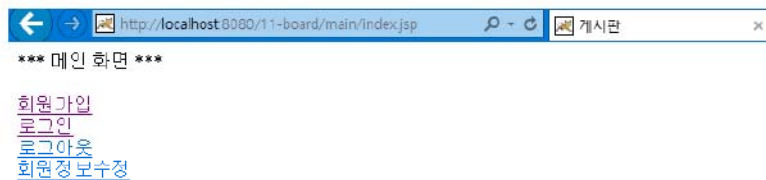


아이디 또는 비밀번호가 틀렸으니 다시 로그인 하세요.



 num1님이 로그인

3. 메인 화면 : 로그인 후



12장. 게시판 2 - 쿠키와 세션

1. Cookie

(1) 쿠키의 개요

- HTTP 프로토콜은 웹 브라우저(클라이언트)의 요청에 대한 응답을 하고 나면 해당 클라이언트와의 연결을 지속하지 않는다.
- 상태가 없는 프로토콜을 위해 상태를 지속시키기 위한 방법이다.
- 정보를 웹 브라우저에 저장한다.



(2) 쿠키의 사용

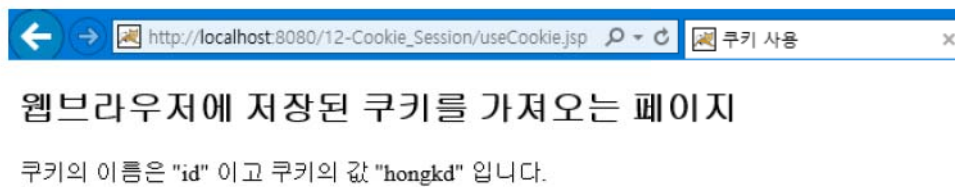
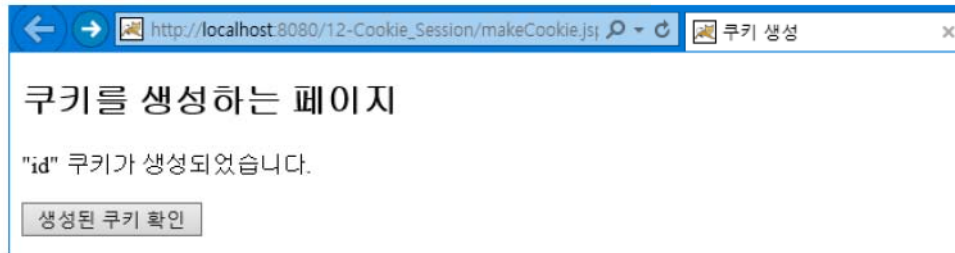
- 이름, 값, 유효기간, 도메인, 경로 등으로 이루어진다.
- 쿠키의 이름은 알파벳과 숫자로 이루어진다.
쿠키 값은 공백, 괄호, 등호, 콤마, 콜론, 세미콜론을 포함하려면 인코딩이 필요하다.
- 쿠키 생성
`Cookie cookie = new Cookie(String name, String value);`
- 쿠키를 response객체에 추가
`response.addCookie(cookie);`
- request객체에 실려 온 쿠키를 읽어 올 때
`Cookie[] cookies = request.getCookies();`
- 쿠키를 작성해서 사용하는 순서
 - ① 먼저 쿠키를 생성한다.
 - ② 쿠키에 필요한 설정을 한다. 쿠키의 유효 시간, 쿠키에 대한 설명 등을 적용하고, 도메인, 패스, 보안등을 한다.
 - ③ 웹 브라우저에 생성된 쿠키를 전송한다.
- 웹 브라우저에 저장된 쿠키를 사용하는 절차
 - ① 웹 브라우저의 요청에서 쿠키를 얻어온다.
 - ② 쿠키는 이름, 값의 쌍으로 된 배열형태로 리턴된다. 리턴 된 쿠키의 배열에서 쿠키 이름을 가져온다.
 - ③ 쿠키 이름을 통해서 해당 쿠키에 설정된 값을 추출한다.

예제1. Cookie 사용하기

Dynamic Web Project : 12-Cookie_Session

JSP File : WebContent/makeCookie.jsp
WebContent/useCookie.jsp

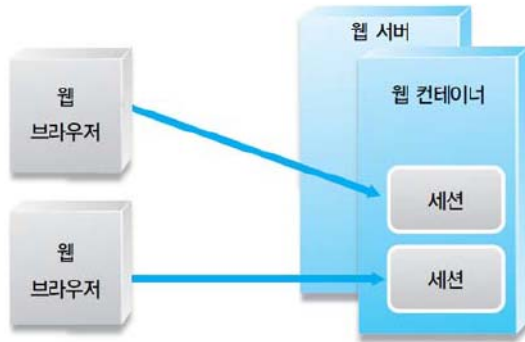
<실행 결과>



2. Session

(1) Session의 개요

- 웹 서버 쪽의 웹 컨테이너에 상태를 유지하기 위한 정보를 저장
 - => 웹 브라우저 당 1개씩 생성되어 웹 컨테이너에 저장
- 웹 브라우저와 웹 서버의 상태 유지가 훨씬 안정적이고 보안상의 문제도 해결
- 웹 서버는 각각의 웹 브라우저로부터 발생한 요청에 대해서 특정한 식별자를 부여
 - => 이것을 사용해서 세션을 구분 및 유지



(2) 세션 사용하기

① 세션의 속성 설정

- => session객체의 `setAttribute()` 메소드를 사용
- => `session.setAttribute("id", "aaaa@king.com");`

② 설정된 세션의 속성을 이용해 정보를 유지

- => session객체의 `getAttribute()` 메소드를 사용
- => `String id = (String)session.getAttribute("id");`

③ 세션의 속성 삭제

- => session객체의 `removeAttribute()` 메소드를 사용
- => `session.removeAttribute("id");`

④ 세션 무효화 - 모든 세션 속성 제거

- => session객체의 `invalidate()` 메소드를 사용
- => `session.invalidate()`

예제2. Session 사용하기

Dynamic Web Project : 12-Cookie_Session

JSP File : WebContent/sessionExamForm.jsp
WebContent/sessionExamPro.jsp

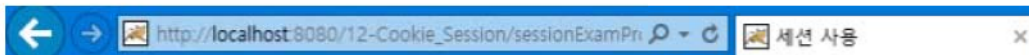
<실행 결과>



정보 입력 폼

아이디 :

패스워드 :



세션 속성 설정 및 사용

id와 passwd 세션 속성을 설정하였습니다.

id속성의 값은 num1이고

passwd속성의 값은 1234입니다.

예제3. 쿠키와 세션으로 게시판 로그인 처리하기

Dynamic Web Project : 11-board

[클라이언트]

Folder : main
JSP File : WebContent/main/index.jsp

Folder : member
JSP File : WebContent/member/loginForm.jsp
WebContent/member/login.jsp (수정)
WebContent/member/loginOk.jsp (수정)
WebContent/member/loginFail.jsp
WebContent/member/logout.jsp (추가)
WebContent/member/writeForm.jsp
WebContent/member/write.jsp
WebContent/member/checkId.jsp

Folder : image
image File : WebContent/image/tumi.png

Folder : sql
sql File : WebContent/sql/member.sql

Folder : script
Javascript File : WebContent/script/memberScript.js

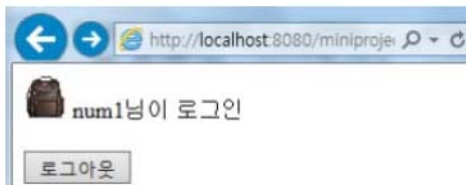
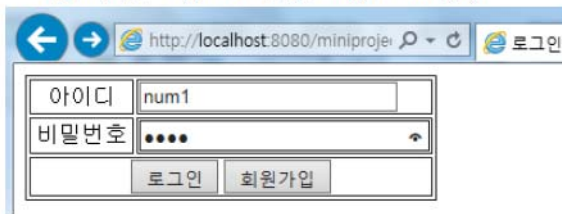
[서버]

Package : member.bean
Class File : src/member.bean/MemberDTO.java
(Data Transfer Object)

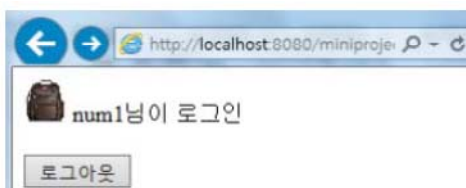
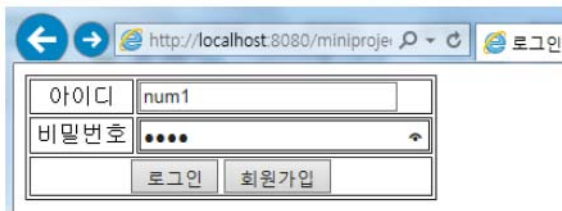
Package : member.dao
Class File : src/member.dao/MemberDAO.java
(Data Access Object)

<실행 결과>

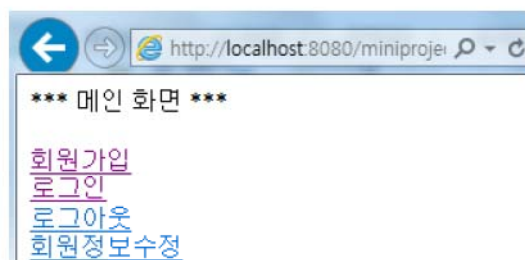
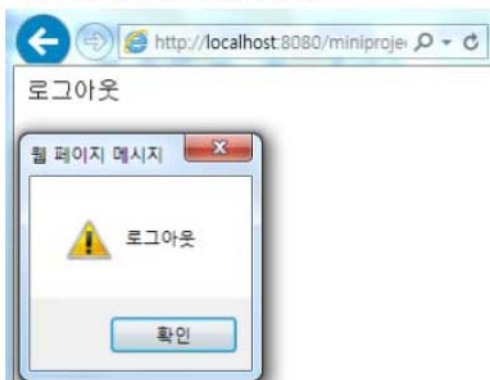
- 쿠키를 이용한 login.jsp 페이지 이동
- login.jsp, loginOk.jsp, loginFail.jsp



- 세션을 이용한 login.jsp 페이지 이동
- login.jsp, loginOk.jsp, loginFail.jsp



- 로그아웃 처리하기
- loginOk.jsp, logout.jsp



13장. 게시판 3 - Connection Pool

1. 시퀀스 객체

- 오라클 객체
- 일정하게 숫자를 증가시켜주는 객체

생성

```
create sequence 시퀀스명  
[increment by 증가값] [start with 시작값]  
[maxvalue 최대값 | minvalue 최소값]  
[cycle | nocycle]  
[cache | nocache]
```

삭제

```
drop sequence 시퀀스객체명;
```

ex)

```
create sequence seq_board nocache nocycle;
```

ex) 1~9 사이에서 2씩 증가됨

```
create sequence test increment by 2 minvalue 1 maxvalue 9 nocache cycle;
```

```
select test.nextval from board;
```

```
drop sequence test;
```

예제1. 게시판에 글쓰기 페이지 추가하기 (글쓰기 폼 페이지만 추가)

Dynamic Web Project : 11-board

[클라이언트]

Folder : main
JSP File : WebContent/main/index.jsp (수정)

Folder : board
JSP File : WebContent/board/boardWriteForm.jsp (추가)
WebContent/board/boardWrite.jsp (추가)

Folder : member
JSP File : WebContent/member/loginForm.jsp
WebContent/member/login.jsp
WebContent/member/loginOk.jsp
WebContent/member/loginFail.jsp
WebContent/member/logout.jsp
WebContent/member/writeForm.jsp
WebContent/member/write.jsp
WebContent/member/checkId.jsp

Folder : image
image File : WebContent/image/tumi.png

Folder : sql
sql File : WebContent/sql/member.sql
WebContent/sql/board.sql (추가)

Folder : script
Javascript File : WebContent/script/memberScript.js
WebContent/script/boardScript.js (추가)

[서버]

Package : member.bean
Class File : src/member.bean/MemberDTO.java
(Data Transfer Object)

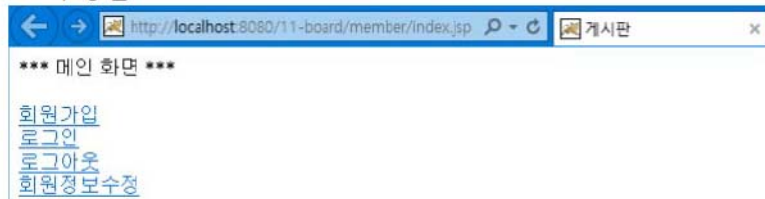
Package : member.dao
Class File : src/member.dao/MemberDAO.java
(Data Access Object)

<실행 결과>

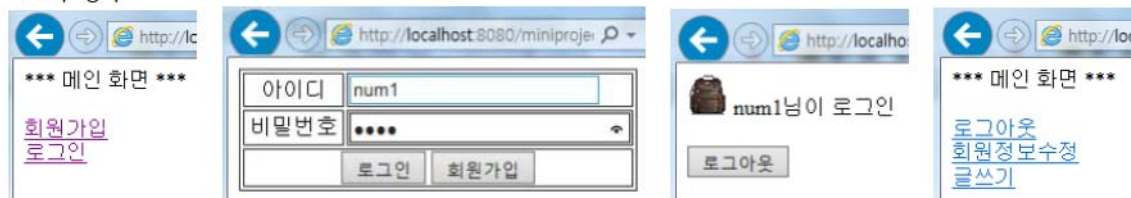
1. 메인 화면 수정

- index.jsp, loginForm.jsp, login.jsp, loginOk.jsp, logout.jsp

=> 수정전

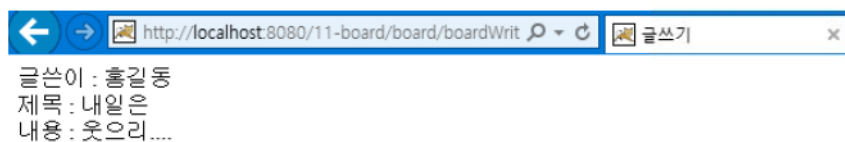
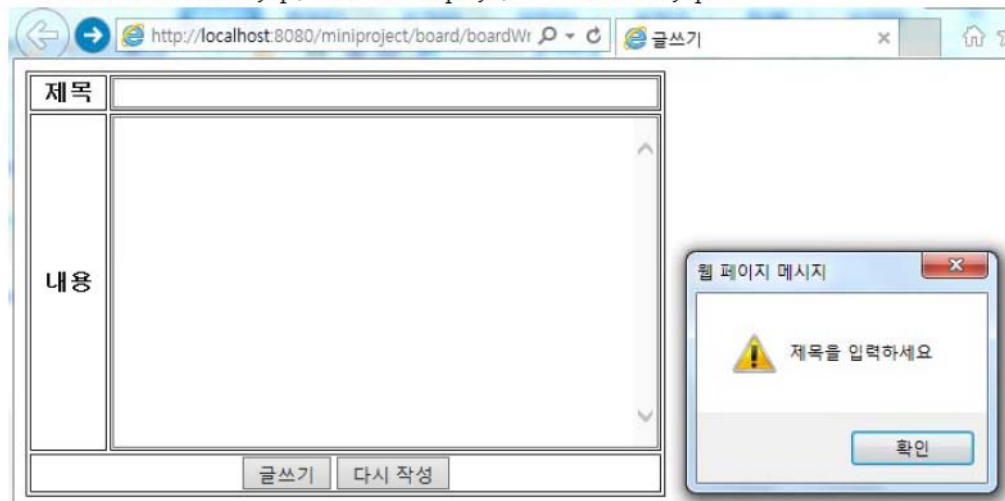


=> 수정후



2. 글쓰기 화면 및 입력 여부 검사하기

- boardWriteForm.jsp, boardScript.js, boardWrite.jsp



예제2. 게시판의 글쓰기 내용을 DB에 저장하기

Dynamic Web Project : 11-board

[클라이언트]

Folder	: main	
JSP File	: WebContent/main/index.jsp	
Folder	: board	
JSP File	: WebContent/board/boardWriteForm.jsp	
	WebContent/board/boardWrite.jsp	(수정)
Folder	: member	
JSP File	: WebContent/member/loginForm.jsp	
	WebContent/member/login.jsp	
	WebContent/member/loginOk.jsp	
	WebContent/member/loginFail.jsp	
	WebContent/member/logout.jsp	
	WebContent/member/writeForm.jsp	
	WebContent/member/write.jsp	
	WebContent/member/checkId.jsp	
Folder	: image	
image File	: WebContent/image/tumi.png	
Folder	: sql	
sql File	: WebContent/sql/member.sql	
	WebContent/sql/board.sql	
Folder	: script	
Javascript File	: WebContent/script/memberScript.js	
	WebContent/script/boardScript.js	

[서버]

Package	: member.bean	
Class File	: src/member.bean/MemberDTO.java	
	src/member.bean/BoardDTO.java	(추가)
Package	: member.dao	
Class File	: src/member.dao/MemberDAO.java	
	src/member.dao/BoardDAO.java	(추가)

<실행 결과>

- sqldeveloper

워크시트 질의 작성기

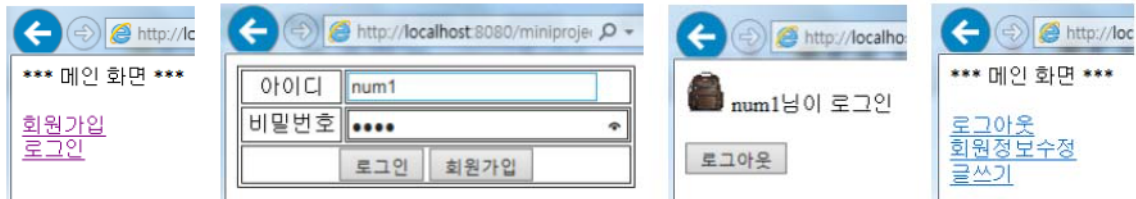
```
-- table 내용 확인
select * from board;
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 1(0.037초)

SEQ	ID	NAME	SUBJECT	CONTENT	HIT	LOGTIME
1	1 num1	홍길동	내일은	웃으리...	0	16/07/03

- 로그인을 먼저 한다. (index.jsp)



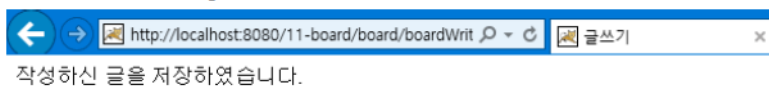
- boardWriteForm.jsp

http://localhost:8080/11-board/board/boardWrit 글쓰기

제목	오늘은....
내용	진짜 답다...

글쓰기 다시작성

- boardWrite.jsp



- sqldeveloper

워크시트 질의 작성기

```
-- table 내용 확인
select * from board;
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 2(0.004초)

SEQ	ID	NAME	SUBJECT	CONTENT	HIT	LOGTIME
1	1 num1	홍길동	내일은	웃으리...	0	16/07/03
2	3 num1	홍길동	오늘은....	진짜 답다...	0	16/07/09

2. 커넥션 풀

* 커넥션 풀의 개요

데이터베이스에 연결하기 위한 커넥션(Connection)은 객체이다.

이 객체는 만들어질 때마다 많은 시스템 자원이 필요하다.

즉, 메모리에 객체를 할당할 자리를 만들고 또 객체가 사용할 여러 자원들에 대한 초기화 작업,

또한 이 객체가 더 이상 필요 없을 때 객체를 거둬들이는 작업이 필요하다.

이런 작업 등이 요구되어서 객체의 생성 작업은 많은 비용을 요구한다.

데이터베이스 커넥션은 데이터베이스에 한번 연결하기 위한 작업인데, 이러한 작업들을 매번 새로운 데이터베이스 연결에 대한 요청이 들어올 때마다 수행해야 한다면 많은 부담이 된다.

이런 문제를 해결하기 위해 커넥션 풀(connection pools)에 커넥션 객체들을 만들어 놓은 후, 커넥션 객체가 필요한 경우 작성한 객체를 할당해 주고, 사용이 끝난 후에는 다시 커넥션 풀로 회수하는 방법을 사용한다.

즉, 한번 만들어져서 사용된 커넥션 객체는 다시 커넥션 풀로 회수되는 것이다.

커넥션 풀은 끊임없이 생성되는 커넥션 문제 해결이 목적으로, 반드시 컨테이너(container)에 1개만 만들어지도록 해야 한다.

* 커넥션

- 데이터베이스 커넥션은 데이터베이스에 한번 연결하기 위한 작업이다.

즉, 어플리케이션에서 데이터베이스에 접속하고, 데이터 처리후 접속을 종료하는 일련의 과정을 커넥션이라 할 수 있다.

- 게시판 목록을 뿌려주는 jsp 페이지에서,

DB에 접속하고 -> SQL 문을 날리고 -> 결과를 받아서 뿌린 후에 -> 연결을 종료한다
이 과정이 커넥션이다.

- 이러한 작업들을 매번 새로운 데이터베이스 연결에 대한 요청이 들어올 때 마다 수행하는 것은 시스템에 과부하를 준다.

* 커넥션 풀

- 커넥션이 들어있는 곳이 바로 Pool(풀) 이다.

- 커넥션 객체들을 만들어 놓은 후, 커넥션 객체가 필요한 경우 작성한 객체를 할당해 주고,
사용이 끝난 후에는 다시 커넥션 풀로 회수하는 방법을 사용한다.

- 사용이 끝나면 close()를 하지 않고, 커넥션을 반납해서, 대기상태로 만드는 것이다.

- 미리 연결해 놓은 커넥션들의 갯수는 조절할 수 있다.

- 오라클 주소, 드라이버, ID, PW를 서버에 숨겨 놓음으로 보안에 좋다.

* 필요한 라이브러리

① [DBCP] : 자카르타(Jakarta) DBCP API 관련 jar 파일

- commons-dbcp-1.4.jar

② [Pool] : 자카르타(Jakarta) DBCP API가 사용하는 자카르타 Pool API의 jar 파일

- commons-pool-1.6.jar

③ [Collections] : Pool API가 사용하는 자카르타 Collection API의 jar 파일

- commons-collections-3.2.1.jar

=> 톰캣에는 기본적으로 tomcat-dbcp.jar가 들어 있다.

그러므로, 따로 위의 라이브러리를 설치할 필요가 없다.

=> C:\Program Files\Apache Software Foundation\Tomcat 8.0\lib\tomcat-dbcp.jar

* 커넥션 풀 사용 방법

① DBCP API관련 jar파일 설치

=> 톰캣에는 기본적으로 tomcat-dbcip.jar가 들어 있다.

② DBCP에 관한 정보 설정

=> context.xml를 만들어서 사용한다.

=> "META-INF"에 만들 => 웹과 상관없는 파일 저장 장소

=> META-INF/context.xml

=> maxActive - 커넥션 풀이 제공할 최대 커넥션 개수

maxIdle - 사용되지 않고 풀에 저장될 수 있는 최대 커넥션의 개수,

음수일 경우 제한을 두지 않음

maxWait - 커넥션 풀을 대기하는 대기시간, 음수일 경우 제한을 두지 않음

③ JSP페이지에서 커넥션 풀 사용

=> 서버의 Connection들을 얻어오려면 javax.sql.DataSource 를 이용

예제3. 게시판의 글쓰기 내용을 커넥션 풀로 DB에 저장하기

Dynamic Web Project : 11-board

[클라이언트]

Folder	: main
JSP File	: WebContent/main/index.jsp
Folder	: board
JSP File	: WebContent/board/boardWriteForm.jsp WebContent/board/boardWrite.jsp
Folder	: member
JSP File	: WebContent/member/loginForm.jsp WebContent/member/login.jsp WebContent/member/loginOk.jsp WebContent/member/loginFail.jsp WebContent/member/logout.jsp WebContent/member/writeForm.jsp WebContent/member/write.jsp WebContent/member/checkId.jsp
Folder	: image
image File	: WebContent/image/tumi.png
Folder	: sql
sql File	: WebContent/sql/member.sql WebContent/sql/board.sql
Folder	: script
Javascript File	: WebContent/script/memberScript.js WebContent/script/boardScript.js

[서버]

Package	: member.bean
Class File	: src/member.bean/MemberDTO.java src/member.bean/BoardDTO.java
Package	: member.dao
Class File	: src/member.dao/MemberDAO.java src/member.dao/BoardDAO.java (수정)
context File	: WebContent/META-INF/context.xml (추가)

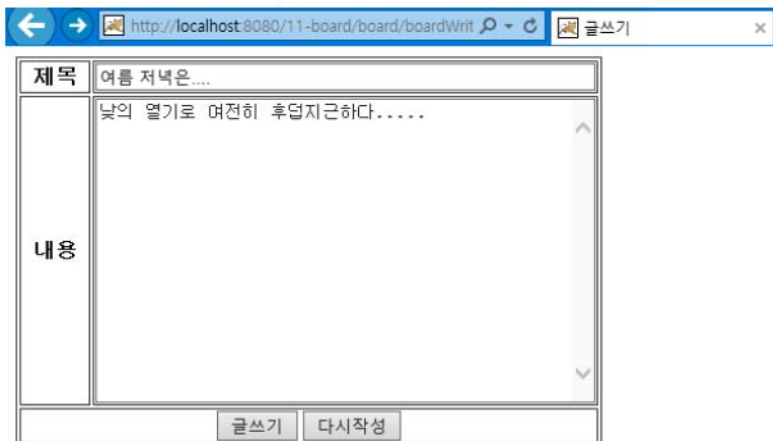
<실행 결과>

- 커넥션 풀로 설정을 변경하여도 동작 자체는 변화가 없다.
- 글쓰기 동작으로 테스트를 한다.
- 앞애와 똑같이 동작되지만 확인하면 된다.

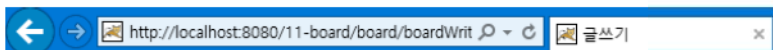
- 로그인을 먼저 한다. (index.jsp)



- boardWriteForm.jsp



- boardWrite.jsp



- sqldeveloper

워크시트

질의 작성기

```
-- table 내용 확인
select * from board;
```

스크립트 출력 x

질의 결과 x

SQL | 인출된 모든 행: 3(0.004초)

SEQ	ID	NAME	SUBJECT	CONTENT	HIT	LOGTIME
1	1 num1	홍길동	내일은	웃으리...	0	16/07/03
2	3 num1	홍길동	오늘은....	진짜 덥다...	0	16/07/09
3	4 num1	홍길동	여름 저녁은....	낮의 열기로 여전히 후덥지근하다.....	0	16/07/09

3. 인덱스 뷰

- 1페이지에 5개씩 게시판 내용물 읽어오기

(1) 인덱스 뷰

-- hidden column - rownum

select * from

(select rownum rn, tt.* from

(select * from board order by seq desc) tt)

where rn>=6 and rn<=10;

select seq, id, name, subject, content, hit, to_char(logtime, 'YYYY.MM.DD') as logtime
from

(select rownum rn, tt.* from

(select * from board order by seq desc) tt)

where rn>=6 and rn<=10;

예제4. 글목록 보기

- 목록 화면 처리하기
=> 1페이지에 5개씩 게시판 내용물 읽어오기
- 목록 항목을 선택하면 내용 보여주기

Dynamic Web Project : 11-board

[클라이언트]

Folder : main
JSP File : **WebContent/main/index.jsp** (수정)

Folder : board
JSP File : WebContent/board/boardWriteForm.jsp
WebContent/board/boardWrite.jsp
WebContent/board/boardView.jsp (추가)
WebContent/board/boardList.jsp (추가)

Folder : member
JSP File : WebContent/member/loginForm.jsp
WebContent/member/login.jsp
WebContent/member/loginOk.jsp
WebContent/member/loginFail.jsp
WebContent/member/logout.jsp
WebContent/member/writeForm.jsp
WebContent/member/write.jsp
WebContent/member/checkId.jsp

Folder : image
image File : WebContent/image/tumi.png

Folder : sql
sql File : WebContent/sql/member.sql
WebContent/sql/board.sql (수정)

Folder : script
Javascript File : WebContent/script/memberScript.js
WebContent/script/boardScript.js

[서버]

Package : member.bean
Class File : src/member.bean/MemberDTO.java
src/member.bean/BoardDTO.java

Package : member.dao
Class File : src/member.dao/MemberDAO.java
src/member.dao/BoardDAO.java (수정)

context File : WebContent/META-INF/context.xml

<실행 결과>

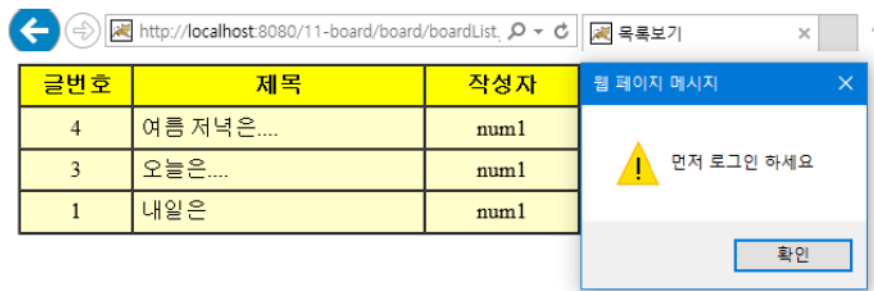
- index.jsp



[회원가입](#)
[로그인](#)
[목록](#)

- boardList.jsp

글번호	제목	작성자	작성일	조회수
4	여름 저녁은....	num1	2016.07.09	0
3	오늘은....	num1	2016.07.09	0
1	내일은	num1	2016.07.03	0



- boardView.jsp

여름 저녁은....		
글번호 : 4	작성자 : 홍길동	조회수 : 0
낮의 열기로 여전히 후덥지근하다.....		

예제5. 글내용 보기 기능 추가

- "목록", "글수정", "글삭제" 버튼 추가하기
- 조회수 증가시키기
- 페이징 처리하기
- 글 삭제

Dynamic Web Project : 11-board

[클라이언트]

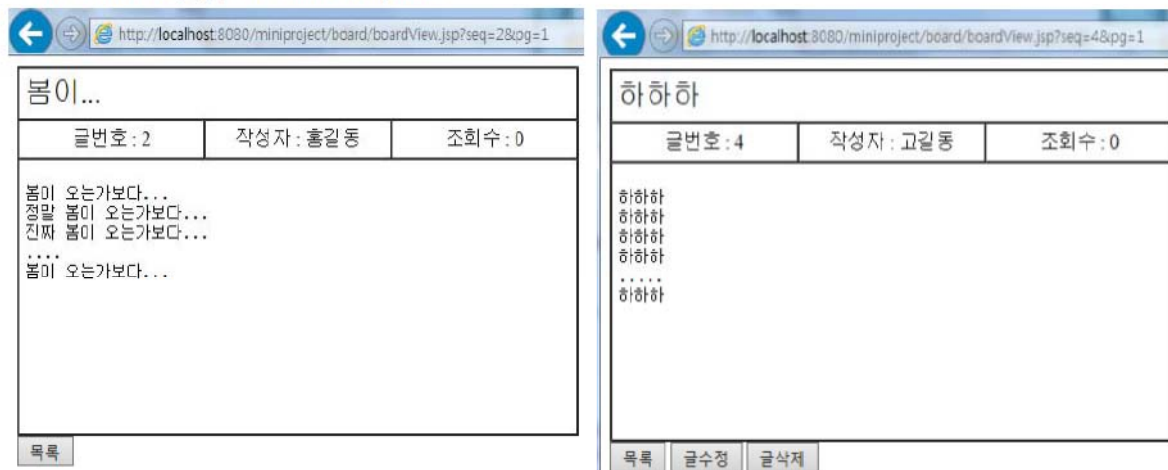
Folder	: main	
JSP File	: WebContent/main/index.jsp	
Folder	: board	
JSP File	: WebContent/board/boardWriteForm.jsp	
	WebContent/board/boardWrite.jsp	(수정)
	WebContent/board/boardView.jsp	(수정)
	WebContent/board/boardList.jsp	(수정)
	WebContent/board/boardDelete.jsp	(추가)
Folder	: member	
JSP File	: WebContent/member/loginForm.jsp	
	WebContent/member/login.jsp	
	WebContent/member/loginOk.jsp	
	WebContent/member/loginFail.jsp	
	WebContent/member/logout.jsp	
	WebContent/member/writeForm.jsp	
	WebContent/member/write.jsp	
	WebContent/member/checkId.jsp	
Folder	: image	
image File	: WebContent/image/tumi.png	
Folder	: sql	
sql File	: WebContent/sql/member.sql	
	WebContent/sql/board.sql	
Folder	: script	
Javascript File	: WebContent/script/memberScript.js	
	WebContent/script/boardScript.js	

[서버]

Package	: member.bean	
Class File	: src/member.bean/MemberDTO.java	
	src/member.bean/BoardDTO.java	
Package	: member.dao	
Class File	: src/member.dao/MemberDAO.java	
	src/member.dao/BoardDAO.java	(수정)
context File	: WebContent/META-INF/context.xml	

<실행 결과>

- 남의 글보기
- 자기 글보기
- boardView.jsp, BoardDAO.java



- 조회수 증가시키기
- boardView.jsp, BoardDAO.java

글번호	제목	작성자	작성일	조회수
4	하하하	num2	2016.03.03	2
3	테스트	num1	2016.03.02	3
2	봄이...	num1	2016.03.02	7
1	봄이...	num1	2016.03.02	0

- 페이징 처리하기
- boardList.jsp, BoardDAO.java

글번호	제목	작성자	작성일	조회수
4	하하하	num2	2016.03.03	2
3	테스트	num1	2016.03.02	3
2	봄이...	num1	2016.03.02	7
1	봄이...	num1	2016.03.02	0

[1] [2] [3]

- 글 삭제
- boardView.jsp, boardDelete.jsp, BoardDAO.java

글번호	제목	작성자	작성일	조회수
14	아...봄이..	num2	2016.03.04	0
13	아...봄이..	num2	2016.03.04	0
12	아...봄이..	num2	2016.03.04	0
11	아...봄이..	num2	2016.03.04	0
10	아...봄이..	num2	2016.03.04	0

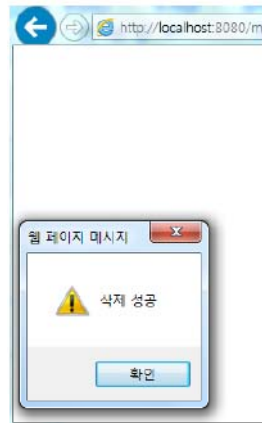
[1] [2] [3]

아...봄아..

글번호 : 14	작성자 : 고길동	조회수 : 1
----------	-----------	---------

벌써 오는거니..
이렇게 오는거니..
겨울이 그리워서니..

목록 글수정 글삭제



글번호	제목	작성자	작성일	조회수
13	아...봄아..	num2	2016.03.04	0
12	아...봄아..	num2	2016.03.04	0
11	아...봄아..	num2	2016.03.04	0
10	아...봄아..	num2	2016.03.04	0
9	아...봄아..	num2	2016.03.04	0

[1] [2] [3]

문제1. 글수정 항목 완성하기

- 로그인 상태에서 글보기 중, 글수정 버튼을 눌러서 내용 수정하고 저장하기

1. BoardDAO.java - 수정

=> boardModify() 메소드 추가

=> boardWrite() 메소드 참조

2. boardScript.js - 수정

=> checkBoardModify() 추가

3. boardModifyForm.jsp - 추가

=> boardWriteForm.jsp 참조

4. boardModify.jsp - 추가

=> boardWrite.jsp 참조

5. boardView.jsp - 수정

=> 글수정 이동 추가

6. sql 문

=> update board set subject='오늘도', content='열심히 살았다...' where seq=19;

문제2 회원정보수정 항목 완성하기

- 메인화면에서 회원정보수정 항목을 눌러서 내용 수정하고 저장하기

1. MemberDAO.java - 수정

=> modify() 메소드 추가

=> write() 메소드 참조

2. memberScript.js - 수정

=> checkModify() 추가

3. modifyForm.jsp - 추가

=> writeForm.jsp 참조

4. modify.jsp - 추가

=> write.jsp 참조

5. index.jsp - 수정

=> 회원정보수정 이동 추가

6. sql 문

=> update member set tel1='010', tel2='2345', tel3='6789' where id='hong';