

8장. spring_step06_jdbc

예제1. jdbc - insert, select, delete

Project Name : step06

package Name : test.service

interface Name : src/main/java/test.service/GoodsService.java

class Name : src/main/java/test.service/GoodsServiceImpl.java

package Name : test.bean

class Name : src/main/java/test.bean/GoodsVO.java

package Name : test.dao

class Name : src/main/java/test.dao/JDBCUtil.java

src/main/java/test.dao/GoodsDAO.java

package Name : test.main

src/main/java/test.main/JdbcExample1.java (main)

src/main/java/test.main/JdbcExample2.java (main)

XML File : src/main/resources/applicationContext.xml

<작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. oracle database 준비하기
5. Class 파일 추가하기
6. XML 파일 만들기

<실행 결과>

- 최종확인은 sqldeveloper에서 한다.

- JdbcExample1.java

==> JDBC로 insertGoods() 기능 처리

==> JDBC로 getGoodsList() 기능 처리

---> GoodsVO [code=p0001, name=Java, price=20000, maker=한샘출판사]

---> GoodsVO [code=p0002, name=JSP 3일완성, price=30000, maker=야메루출판사]

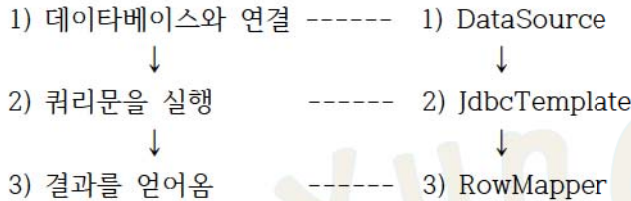
- JdbcExample2.java

==> JDBC로 deleteGoods() 기능 처리

p0001교재가 삭제 되었습니다

9장. spring_step07_spring jdbc

1. 스프링과 JDBC



2. JDBC Template API사용

(1) JdbcTemplate

- 템플릿 메소드 패턴이 적용된 클래스이다.
즉, JDBC의 반복적인 코드를 제거하기 위해 제공되는 클래스.
- 템플릿 메소드 패턴 : 복잡하고 반복되는 알고리즘을 캡슐화해서 재사용하는 패턴
- 반복되는 DB 연동 로직은 JdbcTemplate 클래스의 템플릿 메소드가 제공하고, 개발자는 달라지는 SQL 구문과 설정값만 신경쓰면 된다.

1) update() : INSERT, DELETE, UPDATE SQL문을 수행하기 위해 사용되는 메서드 이다.

ex) `getJdbcTemplate().update("INSERT INTO USER VALUES(?,?,?)",
new Object[]{ name, age, phone});`

2) query() : SELECT문의 실행결과가 여러목록으로 리턴되는 경우에 사용되는 메서드이다

ex) `Object []args = {"%" + searchKeyword + "%"};`

`ArrayList list =`

`getJdbcTemplate().query("SELECT * FROM USERS WHERE USER_NAME LIKE ?",
args, new UserRowMapper());`

3) queryForInt() : SELECT문 실행결과로 리턴되는 정수값을 받기 위해 사용되는 메서드이다.

ex) `int count = getJdbcTemplate().queryForInt("SELECT COUNT(*) FROM USERS");`

4) queryForObject() : SQL수행 결과를 Object로 리턴하는 경우에 이용된다

ex) `String name = getJdbcTemplate().queryForObject(
"SELECT NAME FROM USER WHERE ID=?", new Object[]{"test"}, String.class);`

(2) DataSource 설정

- JdbcTemplate 클래스가 JDBC API를 이용하여 DB연동을 처리하려면 반드시 DB로부터 커넥션을 얻어야 한다.
- DataSource를 <bean> 등록하여 스프링 컨테이너가 생성하도록 해야한다.
- DataSource 인터페이스를 구현한, BasicDataSource 클래스를 사용한다.
- PropertyPlaceholderConfigurer를 이용하면 외부의 프로퍼티 파일을 참조하여, DataSource를 설정할 수 있다.

(3) RowMapper 이용 : SELECT문을 수행하는 결과값을 단순히 정수값이나 문자열로 받지 않고 VO(Value Object)객체에 담아야 하는경우가 있다.

이 경우 VO 객체에 DB에서 SELECT에 데이터가 어떻게 매핑되어야 하는지에 대한 정의가 필요하다. 이 매핑과 관련된 설정을 위해 제공되는 것이 RowMapper 인터페이스이다.

예제1. Spring jdbc - insert, select, delete

Project Name : step07

package Name : test01

interface Name : src/main/java/test01/GoodsService.java

class Name : src/main/java/test01/GoodsVO.java

src/main/java/test01/GoodsRowMapper.java - (추가)

src/main/java/test01/GoodsDAOSpring.java - (추가)

src/main/java/test01/GoodsServiceImpl.java - (수정)

src/main/java/test01/JdbcExample1.java (main)

src/main/java/test01/JdbcExample2.java (main)

XML File : src/main/resources/applicationContext.xml - (추가)

<작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. Class 파일 추가하기
5. XML 파일 만들기

<실행 결과>

- JdbcExample1.java

===> Spring JDBC로 insertGoods() 기능 처리

===> Spring JDBC로 getGoodsList() 기능 처리

---> GoodsVO [code=p0001, name=Java, price=20000, maker=한샘출판사]

---> GoodsVO [code=p0002, name=JSP 3일완성, price=30000, maker=야메루출판사]

- JdbcExample2.java

===> Spring JDBC로 deleteGoods() 기능 처리

p0001교재가 삭제 되었습니다

=> 최종확인은 sqldeveloper에서 한다.

예제2. Spring jdbc - properties 파일을 이용한 DataSource 설정

Project Name : step07

package Name : test01

interface Name : src/main/java/test01/GoodsService.java

class Name : src/main/java/test01/GoodsVO.java

src/main/java/test01/GoodsRowMapper.java

src/main/java/test01/GoodsDAOSpring.java

src/main/java/test01/GoodsServiceImpl.java

src/main/java/test01/JdbcExample1.java (main)

src/main/java/test01/JdbcExample2.java (main)

XML File : [src/main/resources/applicationContext2.xml](#) - (추가)

properties File : [src/main/resources/jdbc.properties](#)

<작업 순서>

1. properties 파일 만들기
2. XML 파일 만들기

<실행 결과>

- JdbcExample1.java

===> Spring JDBC로 insertGoods() 기능 처리

===> Spring JDBC로 getGoodsList() 기능 처리

---> GoodsVO [code=p0001, name=Java, price=20000, maker=한샘출판사]

---> GoodsVO [code=p0002, name=JSP 3일완성, price=30000, maker=야메루출판사]

- JdbcExample2.java

===> Spring JDBC로 deleteGoods() 기능 처리

p0001교재가 삭제 되었습니다

=> 최종확인은 sqldeveloper에서 한다.

10장. spring_step08_myBatis

1. MyBatis

- site : <http://blog.mybatis.org>
- 다운로드 : mybatis-3.4.2.zip

* MyBatis

- iBATIS(아이바티스)는 SQL에 기반한 데이터베이스와 자바, 닷넷(.NET), 루비(Ruby) 등을 연결시켜 주는 역할을 하는 영속성 프레임워크(Persistence Framework)이다.
이러한 연결은 프로그램의 소스코드에서 SQL 문장을 분리하여 별도의 XML 파일로 저장하고, 이 둘을 서로 연결시켜주는 방식으로 작동한다.
- iBatis는 사용자가 SQL 문장을 만들면 그에 적합한 객체모델을 생성하는 방식으로 작동한다.
- 구글 코드에서 새로이 만든 iBATIS 프레임워크의 이름이 MyBatis로 변경되었다.
- MyBatis의 가장 중요한 특징을 두가지로 정리하면,
=> 한두 줄의 자바코드로 DB 연동을 처리한다.
=> SQL 명령어를 자바코드에서 분리하여 XML파일에 따로 관리한다.

ex) 다음의 매핑을 MYBATIS XML 매핑 파일에 추가한다:

```
<select id="getProduct" parameterClass="java.lang.Long" resultClass="com.example.Product">
    select PROD_ID as id, PROD_DESC as description
    from PRODUCT
    where PROD_ID = #value#
</select>
```

그러면 제품 번호 123에 대해 데이터베이스에서 새로운 자바 Product 오브젝트를 다음과 같이 검색한다:

```
Product resultProduct = (Product) sqlMapClient.queryForObject("getProduct", 123);
```

* SqlSession

- MyBatis를 스프링과 함께 사용하려면 SqlSession을 정의해야 한다.
- MyBatis에서는 SqlSession를 생성하기 위해 SqlSessionFactory를 사용한다.
- 세션을 한번 생성하면 매핑구문을 실행하거나 커밋 또는 롤백을 하기 위해 세션을 사용할 수 있다.
- 마지막으로 더 이상 필요하지 않은 상태가 되면 세션을 닫는다.
- MyBatis 스프링 연동모듈을 사용하면 SqlSessionFactory를 직접 사용할 필요가 없다.
왜냐하면, 스프링 트랜잭션 설정에 따라 자동으로 커밋 혹은 롤백을 수행하고 닫혀지는 스레드에 안전한 SqlSession 개체가 스프링 빈에 주입될 수 있기 때문이다.

예제1. jdbc - myBatis

Project Name : step08
package Name : test01
interface Name : src/main/java/test01/GoodsService.java
class Name : src/main/java/test01/GoodsVO.java
src/main/java/test01/SqlMapClientFactory.java
src/main/java/test01/GoodsDAO.java
src/main/java/test01/GoodsServiceImpl.java
src/main/java/test01/JdbcExample1.java (main)
src/main/java/test01/JdbcExample2.java (main)
XML File : src/main/resources/bean.xml
src/main/resources/mybatis-config.xml
src/main/resources/goods-mapping.xml
properties File : src/main/resources/db.properties

<작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. Class 파일 추가하기
5. properties 파일 만들기
6. XML 파일 만들기
7. mybatis XML 코딩

<실행 결과>

=> 최종확인은 sqldeveloper에서 한다.

- JdbcExample1.java

```
Console Progress Problems
<terminated> JdbcExample1 (3) [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (2017. 11. 17. 오후 8:34:26)
11월 17, 2017 8:34:29 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [bean.xml]
11월 17, 2017 8:34:29 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@377dca04: startup date [Fri Nov 17 20:34:29.268 [main] DEBUG org.apache.ibatis.logging.LogFactory - Logging initialized using 'class org.apache.ibatis.logg
20:34:29.393 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
20:34:29.393 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
20:34:29.393 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
20:34:29.393 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
====> Mybatis로 insertGoods() 기능 처리
20:34:29.486 [main] DEBUG o.a.i.t.jdbc.JdbcTransaction - Opening JDBC Connection
20:34:30.190 [main] DEBUG o.a.i.d.pooled.PooledDataSource - Created connection 1666607455.
20:34:30.190 [main] DEBUG o.a.i.t.jdbc.JdbcTransaction - Setting autocommit to false on JDBC Connection [oracle.jdbc.driv
20:34:30.192 [main] DEBUG mybatis.goodsMapper.insertGoods - ==> Preparing: INSERT INTO GOODS (CODE, NAME, PRICE, MAKER)
20:34:30.302 [main] DEBUG mybatis.goodsMapper.insertGoods - ==> Parameters: p0001(String), Java(String), 20000(Integer),
20:34:30.302 [main] DEBUG mybatis.goodsMapper.insertGoods - <== Updates: 1
====> Mybatis로 getGoodsList() 기능 처리
20:34:30.302 [main] DEBUG mybatis.goodsMapper.getGoodsList - ==> Preparing: SELECT * FROM GOODS
20:34:30.302 [main] DEBUG mybatis.goodsMapper.getGoodsList - ==> Parameters:
20:34:30.364 [main] DEBUG mybatis.goodsMapper.getGoodsList - <== Total: 2
---> GoodsVO [code=p0002, name=JSP 3일완성, price=30000, maker=아메리칸출판사]
---> GoodsVO [code=p0001, name=Java, price=20000, maker=한샘출판사]
11월 17, 2017 8:34:30 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@377dca04: startup date [Fri Nov 17 20:34:29
```

- JdbcExample2.java

```
Console Progress Problems
<terminated> JdbcExample2 (3) [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (2017. 11. 17. 오후 8:33:32)
11월 17, 2017 8:33:32 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [bean.xml]
11월 17, 2017 8:33:32 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@377dca04: startup date [Fri Nov 17 20:3
20:33:32.977 [main] DEBUG org.apache.ibatis.logging.LogFactory - Logging initialized using 'class org.apache.ibatis.logg
20:33:33.086 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
20:33:33.086 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
20:33:33.086 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
20:33:33.086 [main] DEBUG o.a.i.d.pooled.PooledDataSource - PooledDataSource forcefully closed/removed all connections.
====> Mybatis로 deleteGoods() 가능 처리
20:33:33.180 [main] DEBUG o.a.i.t.jdbc.JdbcTransaction - Opening JDBC Connection
20:33:33.868 [main] DEBUG o.a.i.d.pooled.PooledDataSource - Created connection 1666607455.
20:33:33.868 [main] DEBUG o.a.i.t.jdbc.JdbcTransaction - Setting autocommit to false on JDBC Connection [oracle.jdbc.dri
20:33:33.874 [main] DEBUG mybatis.goodsMapper.deleteGoods - ==> Preparing: DELETE FROM GOODS WHERE CODE=?
20:33:33.984 [main] DEBUG mybatis.goodsMapper.deleteGoods - ==> Parameters: p0001(String)
20:33:33.999 [main] DEBUG mybatis.goodsMapper.deleteGoods - <== Updates: 0
p0001코드가 삭제 되었습니다
11월 17, 2017 8:33:33 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@377dca04: startup date [Fri Nov 17 20:33:3
```


11장. spring_step09_model 1 (1)

1. Tomcat 서버 설치 및 STS 연결

(1) Tomcat 서버 설치

1) Tomcat

- Servlet이나 JSP의 컨테이너
- tomcat.apache.org
 - => Download/Tomcat 8.0
 - => apache-tomcat-8.0.45-windows-x64.zip 클릭
 - => 작업폴더에 저장

2) 설치

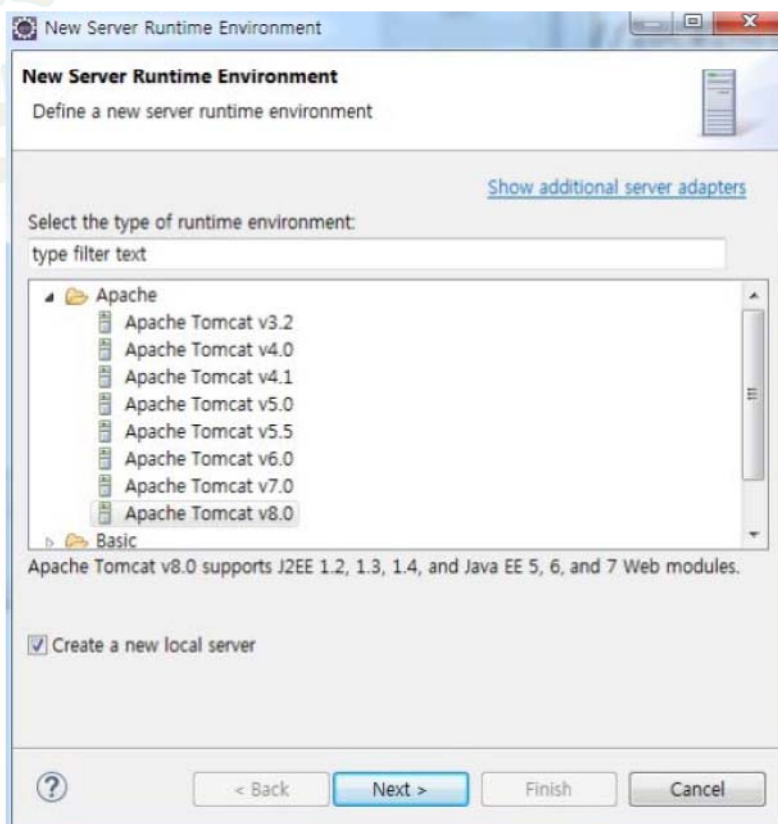
- spring 폴더에 압축을 푼다.
- apache-tomcat-8.0.45 이름으로 폴더가 생긴다.

(2) STS에 Tomcat 연결하기

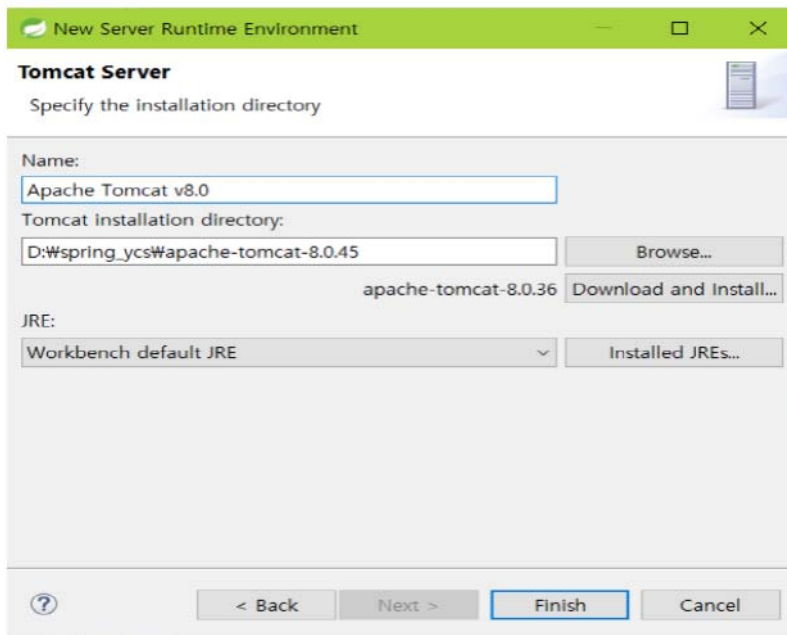
1) 메뉴 -> Window -> Preferences 선택

2) Server -> Runtime Environments 선택

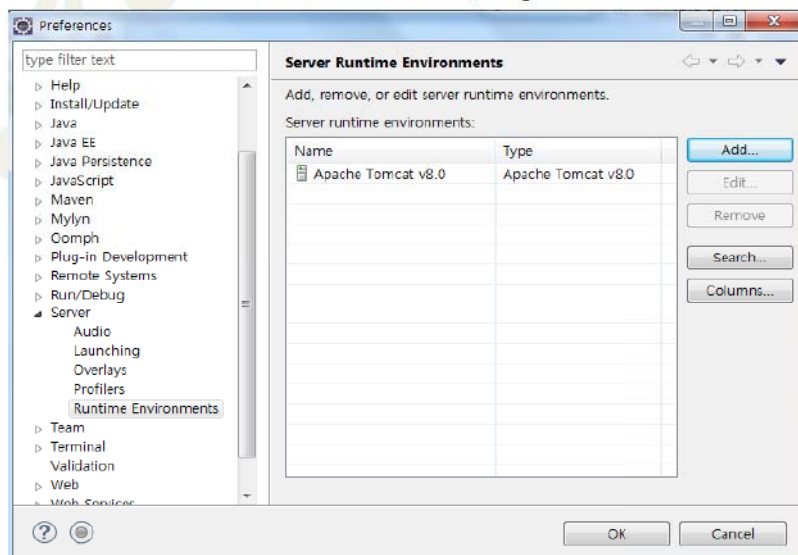
- => "Server Runtime Environment"에서 "Add" 버튼 클릭
- => "New Server Runtime Environment" 창에서
 - > "Apache/Apache Tomcat v8.0" 선택 -> "Next" 버튼 클릭



- => "Tomcat Server" 창에서
 - > "Browse" 버튼 클릭
 - > "D:\spring_ycs\apache-tomcat-8.0.45" 선택
 - > "Finish" 버튼 클릭



=> "Server Runtime Environment"에 "Apache Tomcat v8.0"이 추가된다.

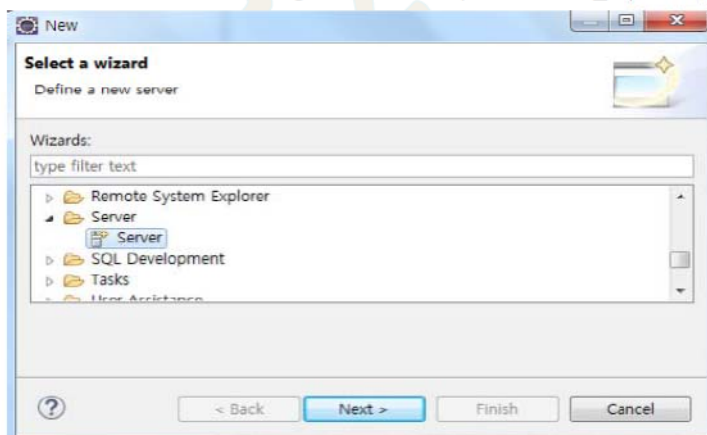


=> "OK" 버튼 클릭

3) Server 추가하기 (Server가 없을 때)

=> "Project Explorer" -> 오-마 -> New -> Other 선택

=> "Select a wizard" 창 -> "Server/Server" 선택 -> "Next" 버튼 클릭



=> "Finish" 버튼 클릭

=> "Project Explorer"에 "Servers" 가 추가된다.

2. 로그인 및 게시판

예제1. 회원 DB 만들기

- * Database 계정에 테이블 만들기
 - sqldeveloper 실행하기
 - edu/1234 연결하기
 - member 테이블 만들기
 - 데이터 추가, 검색, 수정, 삭제

① 테이블 작성 및 삭제

-- 테이블 생성

```
create table member(  
name varchar2(30) not null,  
id varchar2(30) primary key,    --기본키, unique, not null - 무결성 제약 조건  
pwd varchar2(30) not null,  
gender varchar2(3),  
email1 varchar2(20),  
email2 varchar2(20),  
tel1 varchar2(10),  
tel2 varchar2(10),  
tel3 varchar2(10),  
addr varchar2(100),  
logtime date);
```

-- 테이블 구조 확인

desc member;

-- 테이블 삭제

drop table member purge;

② 데이터 추가, 검색, 수정, 삭제

-- 데이터 추가

```
insert into member values ('홍길동', 'num1', '1234', '남', 'hong', 'naver.com', '010',  
                           '1234', '5678', '경기도 수원시', sysdate);
```

-- 데이터 검색

select * from member;

select * from member where name='홍길동';

-- 데이터 수정

update member set tel2='2345' where id='num1';

-- 데이터 삭제

delete member where id = 'num1';

-- 파일 저장

commit;

예제2. 로그인 처리하기

Dynamic Web Project : step09

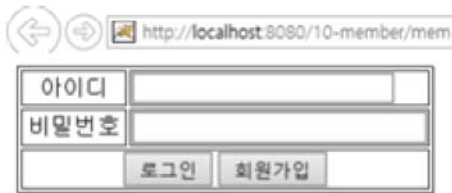
[클라이언트]

Folder : member
JSP File : WebContent/member/loginForm.jsp
JSP File : WebContent/member/login.jsp
Package : member.dao
Class File : src/member.dao/MemberDAO.java
(Data Access Object)

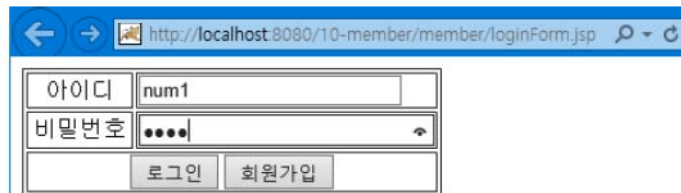
<작업 순서>

1. 프로젝트 만들기
2. 라이브러리 추가하기
3. loginForm.jsp
4. MemberDAO.java
5. login.jsp

<실행 결과>



아이디	<input type="text"/>
비밀번호	<input type="password"/>
	<input type="button" value="로그인"/> <input type="button" value="회원가입"/>



아이디	num1
비밀번호
	<input type="button" value="로그인"/> <input type="button" value="회원가입"/>

예제3. 게시판 목록보기

- 목록 화면 처리하기
 - => 1페이지에 5개씩 게시판 내용물 읽어오기

Dynamic Web Project : step09

Package : member.dao
Class File : src/member.dao/MemberDAO.java
Package : board.bean
Class File : src/board.bean/BoardDTO.java
Package : board.dao
Class File : src/board.dao/BoardDAO.java

Folder : member
JSP File : WebContent/member/loginForm.jsp
WebContent/member/login.jsp

Folder : board
JSP File : WebContent/board/boardList.jsp

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

<작업 순서>

- ① BoardDTO.java
 - 회원 정보를 저장시킬 Java Beans 클래스 만들기
- ② BoardDAO.java
 - DB의 데이터 처리하기
- ③ boardList.jsp
 - 로그인되어 있을 때, 선택된 글 내용 보여주기

<실행 결과>

- boardList.jsp



글번호	제목	작성자	작성일	조회수
3	내일은	num1	2017.11.21	0
1	내일은	num1	2017.11.20	0

[1]

[새글쓰기](#)

예제4. 글쓰기 추가, 글내용 보기, 글 삭제

- 목록 항목을 선택하면 내용 보여주기
- 글 삭제

Dynamic Web Project : step09

Package	: member.dao
Class File	: src/member.dao/MemberDAO.java
Package	: board.bean
Class File	: src/board.bean/BoardDTO.java
Package	: board.dao
Class File	: src/board.dao/BoardDAO.java
Folder	: member
JSP File	: WebContent/member/loginForm.jsp WebContent/member/login.jsp
Folder	: board
JSP File	: WebContent/board/boardWriteForm.jsp WebContent/board/boardWrite.jsp WebContent/board/boardDelete.jsp WebContent/board/boardView.jsp WebContent/board/boardList.jsp
Folder	: sql
sql File	: WebContent/sql/member.sql WebContent/sql/board.sql
Folder	: script
Javascript File	: WebContent/script/boardScript.js
context File	: WebContent/META-INF/context.xml

<작업 순서>

- ① boardWriteForm.jsp
 - 글쓰기 화면
- ② boardScript.js
 - 유효성 검사
- ③ boardWrite.jsp
 - 글쓰기 처리
- ④ boardView.jsp
 - 목록 보여주기 화면 구성
 - 데이터베이스에서 목록 내용을 전달받아서 화면에 보여주기

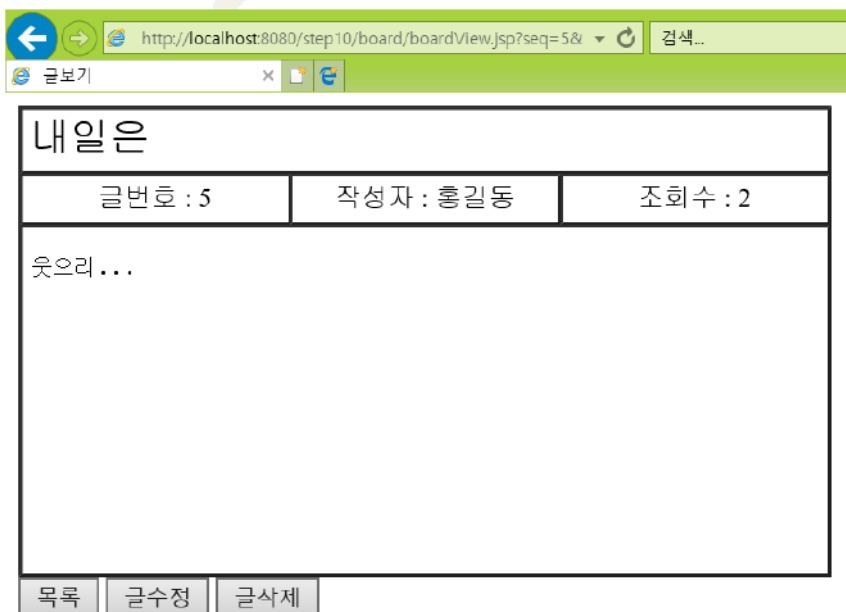
<실행 결과>

1. 글쓰기 화면 및 입력 여부 검사하기



글번호	제목	작성자	작성일	조회수
5	내일은	num1	2017.11.23	1
4	오늘은	num1	2017.11.21	1
1	내일은	num1	2017.11.20	1

[새글쓰기](#)

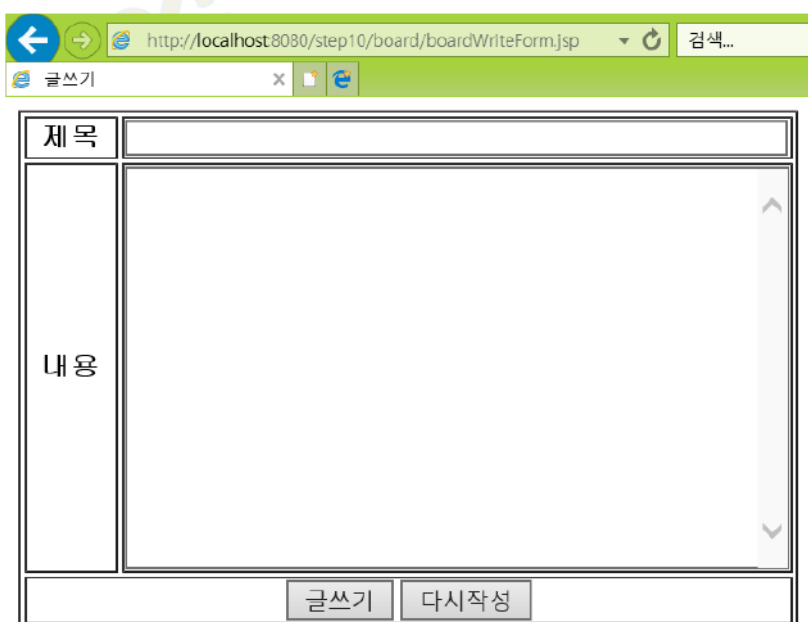


내일은

글번호 : 5	작성자 : 홍길동	조회수 : 2
---------	-----------	---------

웃으라...

목록 글수정 글삭제



제목	
내용	

글쓰기 다시작성

12장. spring_step10_model 1 (2)

예제1. Maven으로 프로젝트 구성

Dynamic Web Project : step10

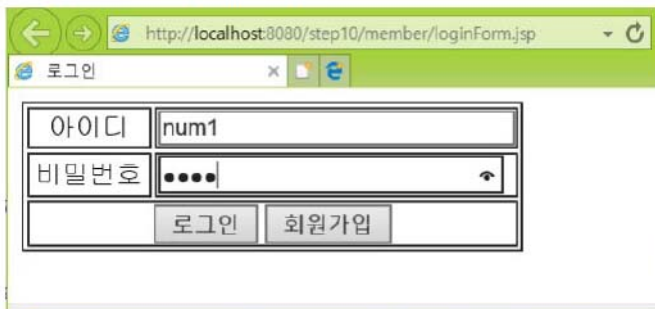
Package	: member.dao
Class File	: src/main/java/member.dao/MemberDAO.java
Package	: board.bean
Class File	: src/main/java/board.bean/BoardDTO.java
Package	: board.dao
Class File	: src/main/java/board.dao/BoardDAO.java
Folder	: member
JSP File	: src/main/webapp/member/loginForm.jsp src/main/webapp/member/login.jsp
Folder	: board
JSP File	: src/main/webapp/board/boardWriteForm.jsp src/main/webapp/board/boardWrite.jsp src/main/webapp/board/boardDelete.jsp src/main/webapp/board/boardView.jsp src/main/webapp/board/boardList.jsp
Folder	: sql
sql File	: src/main/webapp/sql/member.sql src/main/webapp/sql/board.sql
Folder	: script
Javascript File	: src/main/webapp/script/boardScript.js

<작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. pom.xml 파일에 추가하기
5. web.xml에 환경설정하기
6. Class 파일 추가하기
7. jsp 파일 추가하기

<실행 결과>

- loginForm.jsp에서 시작




글번호	제목	작성자	작성일	조회수
5	내일은	num1	2017.11.23	1
4	오늘은	num1	2017.11.21	1
1	내일은	num1	2017.11.20	1

[1]

[새글쓰기](#)

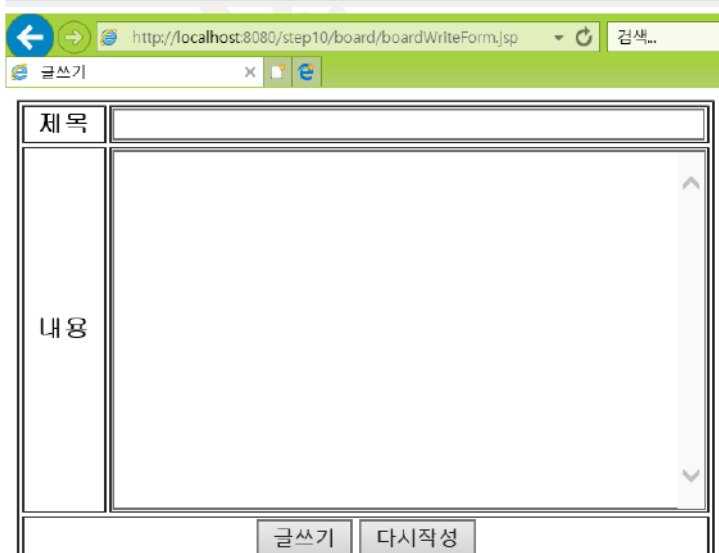


내일은

글번호 : 5	작성자 : 홍길동	조회수 : 2
---------	-----------	---------

웃으리...

목록 글수정 글삭제



13장. spring_step11_model 2

- Model 2 아키텍처 = MVC 아키텍처
- Model 2의 가장 중요한 특징은 Controller의 사용이다.
- 기존에 JSP가 담당했던 Controller 로직이 별도의 Controller 기능의 서블릿으로 옮겨졌다.
즉, 기존 JSP의 자바코드와 HTML코드 중에서 자바코드를 분리시킨 것이다.
자바코드 => Controller => 서블릿 => 웹 프로그래머
HTML코드 => View => JSP => 웹 디자이너
- MVC (Model & View & Controller)

기능	구성
Model	DTO, DAO class
View	JSP file
Controller	Servlet class

예제1. Controller로 프로젝트 구성

Dynamic Web Project : step11

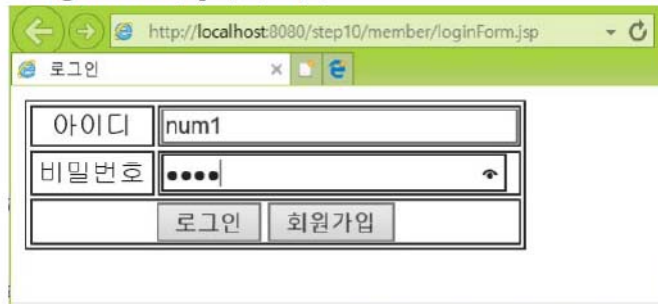
Package	: member.dao
Class File	: src/main/java/member.dao/MemberDAO.java
Package	: board.bean
Class File	: src/main/java/board.bean/BoardDTO.java
Package	: board.dao
Class File	: src/main/java/board.dao/BoardDAO.java
Package	: board.controller
Servlet File	: src/main/java/board.controller/DispatcherServlet.java
Folder	: member
JSP File	: src/main/webapp/member/loginForm.jsp
Folder	: board
JSP File	: src/main/webapp/board/boardWriteForm.jsp src/main/webapp/board/boardWrite.jsp src/main/webapp/board/boardDelete.jsp src/main/webapp/board/boardView.jsp src/main/webapp/board/boardList.jsp
Folder	: sql
sql File	: src/main/webapp/sql/member.sql src/main/webapp/sql/board.sql
Folder	: script
Javascript File	: src/main/webapp/script/boardScript.js

<작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. pom.xml 파일에 추가하기
5. web.xml에 환경설정하기
6. Servlet 파일 추가하기
7. Class 파일 추가하기
8. jsp 파일 추가하기

<실행 결과>

- loginForm.jsp에서 시작




글번호	제목	작성자	작성일	조회수
5	내일은	num1	2017.11.23	1
4	오늘은	num1	2017.11.21	1
1	내일은	num1	2017.11.20	1

[\[1\]](#)

[새글쓰기](#)

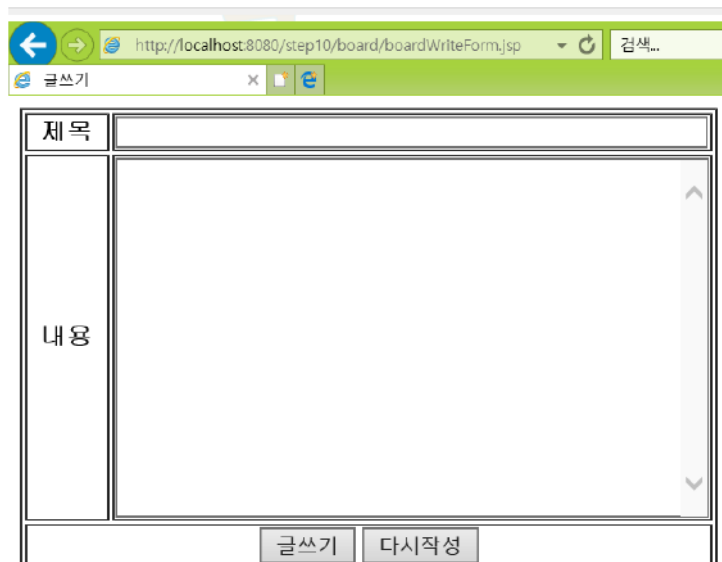


내일은

글번호 : 5	작성자 : 홍길동	조회수 : 2
---------	-----------	---------

웃으리...

목록 글수정 글삭제



제목

내용

글쓰기 다시작성