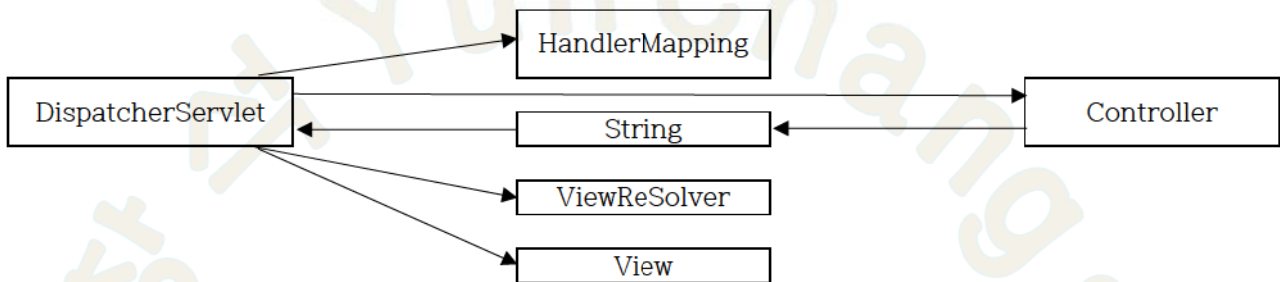


## 14장. spring\_step12\_MVC (1)

- DispatcherServlet 클래스 하나로 Controller 기능을 구현하면, 수많은 분기처리 로직을 처리하기 때문에 개발 및 유지 보수가 힘들게 된다.
- MVC 프레임워크를 사용하면, MVC 프레임워크에서 제공하는 Controller를 사용하게 된다. 그러면, 직접 우리가 Controller 기능을 구현하지 않아도 된다.
- MVC 프레임워크를 사용하는 이유는 효율적인 Controller를 제공하기 때문이다.
- MVC 프레임워크 구조



클래스	기능
DispatcherServlet	유일한 서블릿 클래스. 모든 클라이언트의 요청을 가장 먼저 처리하는 front controller
HandlerMapping	클라이언트의 요청을 처리할 controller 매핑
Controller	실질적인 클라이언트의 요청 처리
ViewReSolver	controller가 리턴한 view 이름으로 실행될 jsp 경로 완성
View	결과를 출력할 JSP 파일

## 예제1. MVC 프레임워크 구현해보기

### Dynamic Web Project : step12

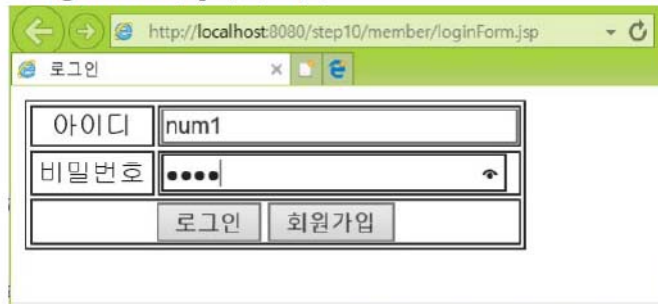
Package	: member.dao
Class File	: src/main/java/member.dao/MemberDAO.java
Package	: board.bean
Class File	: src/main/java/board.bean/BoardDTO.java
Package	: board.dao
Class File	: src/main/java/board.dao/BoardDAO.java
Package	: board.controller
Servlet File	: src/main/java/board.controller/DispatcherServlet.java
Interface File	: src/main/java/board.controller/Controller.java
Class File	: src/main/java/board.controller/HandlerMapping.java
	src/main/java/board.controller/ViewResolver.java
	src/main/java/board.controller/LoginController.java
	src/main/java/board.controller/BoardListController.java
	src/main/java/board.controller/BoardViewController.java
	src/main/java/board.controller/BoardWriteController.java
	src/main/java/board.controller/BoardDeleteController
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
9. Controller 인터페이스 만들기
10. LoginController 클래스 구현하기
11. HandlerMapping 클래스 만들기
12. ViewResolver 클래스 만들기
13. 나머지 Controller도 만들기

## <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](#)

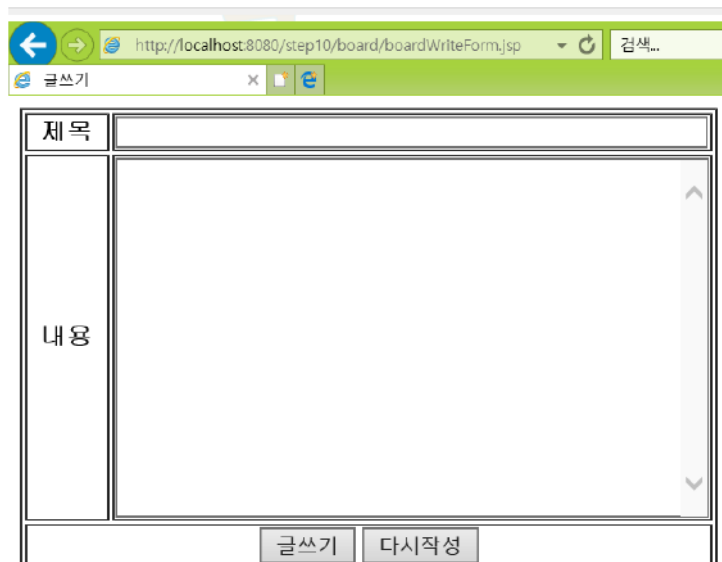


내일은

글번호 : 5      작성자 : 홍길동      조회수 : 2

웃으리...

목록    글수정    글삭제



제목

내용

글쓰기    다시작성

## 15장. spring\_step13\_MVC (2)

### 예제1. jsp에 EL/JSTL 적용하기

#### Dynamic Web Project : step13

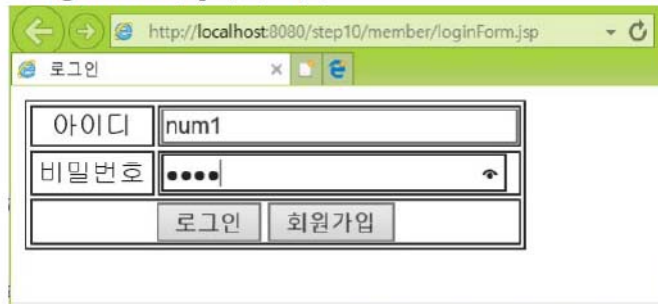
Package	: member.dao
Class File	: src/main/java/member.dao/MemberDAO.java
Package	: board.bean
Class File	: src/main/java/board.bean/BoardDTO.java
Package	: board.dao
Class File	: src/main/java/board.dao/BoardDAO.java
Package	: board.controller
Servlet File	: src/main/java/board.controller/DispatcherServlet.java
Interface File	: src/main/java/board.controller/Controller.java
Class File	: src/main/java/board.controller/HandlerMapping.java src/main/java/board.controller/ViewResolver.java src/main/java/board.controller/LoginController.java src/main/java/board.controller/BoardListController.java src/main/java/board.controller/BoardViewController.java src/main/java/board.controller/BoardWriteController.java src/main/java/board.controller/BoardDeleteController
Folder	: member
JSP File	: src/main/webapp/member/loginForm.jsp
Folder	: board
JSP File	: src/main/webapp/board/boardWriteForm.jsp src/main/webapp/board/boardDelete.jsp <b>src/main/webapp/board/boardWrite.jsp</b> <b>src/main/webapp/board/boardView.jsp</b> <b>src/main/webapp/board/boardList.jsp</b>
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](#)

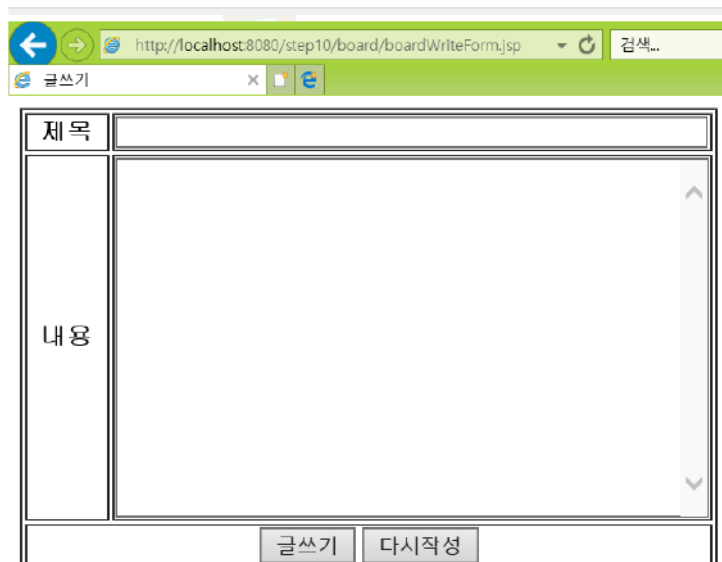


내일은

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

웃으리...

목록 글수정 글삭제



제목

내용

글쓰기 다시작성



## 16장. spring\_step14\_Spring MVC (1)

### 1. Spring MVC

#### (1) 스프링 MVC 개요

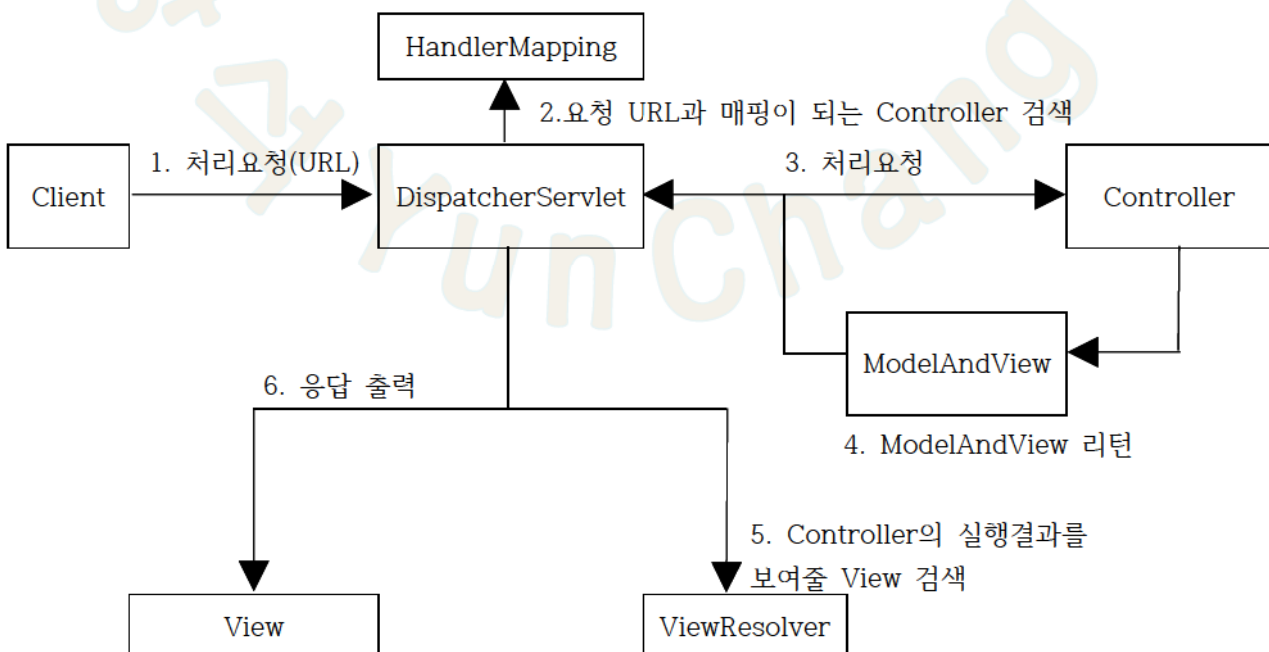
- DI or AOP 등을 쉽게 사용
- 추가 설정을 하지 않아도 된다는 장점을 가짐.

#### (2) 장점

- 스프링 MVC 프레임워크는 스프링 기반으로 사용할 수 있다.
- 스프링이 제공하는 트랜잭션처리가 DI 및 AOP 적용 등을 손쉽게 사용할 수 있다.
- 스트럿츠와 같은 프레임워크와 스프링 프레임워크를 연동하기 위해 추가적인 설정을 하지 않아도 된다.

#### (3) 스프링MVC 주요 구성 요소

구성요소	설 명
DispatcherServlet	클라이언트의 요청을 전달 받는다. 컨트롤러에게 클라이언트의 요청을 전달하고 컨트롤러가 리턴한 결과값을 View에 전달하여 알맞은 응답을 생성하도록 한다. (웹브라우저로부터 송신된 Request를 일괄적으로 관리한다.)
HandlerMapping	클라이언트의 요청URL을 어떤 컨트롤러가 처리할 지를 결정한다. RequestURL과 Controller 클래스의 매핑을 관리한다. 즉, DispatcherServlet에서 바로 Controller로 보내는게 아니라 handlerMapping을 통해 Requesturl을 구분해서 Controller로 전달한다.)
Controller	클라이언트의 요청을 처리한 뒤, 그 결과를 DispatcherServlet에 알려준다. 스트럿츠의 Action과 동일한 역할을 수행한다. 비즈니스 로직을 호출하여 처리 결과 ModelAndView 인스턴스를 반환한다
ModelAndView	컨트롤러가 처리한 결과 정보 및 뷰 선택에 필요한 정보를 담는다. 즉, 데이터와 넘길 페이지의 값을 모두 가지고 있는 객체이다.
ViewResolver	컨트롤러의 처리결과로 생성할 뷰를 결정한다. Controller 클래스로부터 반환된 View 정보가 논리적인 View이름일 경우에는 bean설정 파일에 정의되어있는 ViewResolver 클래스를 이용하여 클라이언트에게 출력할 View 객체를 얻게 된다.
View	컨트롤러의 처리결과 화면을 생성한다.



(4) 웹 어플리케이션을 개발하는 5단계

- ① 클라이언트의 요청을 받을 DispatcherServlet을 web.xml에 설정한다.
- ② Handlermapping을 이용하여 요청URL과 컨트롤러를 작성한다.
- ③ 클라이언트의 요청을 처리할 컨트롤러를 작성한다.
- ④ 어떤 뷰를 이용하여 컨트롤러의 처리결과 응답화면을 생성할지 결정하는 ViewResolver를 설정한다.
- ⑤ JSP를 이용하여 뷰 영역의 코드를 작성한다.

(5) Annotation

- @Component  
=> 컴포넌트클래스
- @Controller  
=> mvc에서 컨트롤러 클래스
- @Service  
=> Service Layer의 클래스
- @Repository  
=> Data Access Object 클래스
- @Autowired  
=> Autowired 어노테이션이 붙은 인스턴스 변수는 해당 변수 타입과 일치하는 컨텍스트 내의 빈을 찾아 인스턴스 변수에 주입해 준다.  
의존성 주입을 위해선 생성자나 setter가 필요한데 이 어노테이션을 사용할 경우 없어도 가능하다

(6) pom.xml을 사용하지 않고, 라이브러리 다운받아 추가할 경우

WebContent

WEB-INF

```
lib -- 다운받은 라이브러리 추가할 것
commons-logging-1.1.3.jar
spring-beans-3.2.3.RELEASE.jar
spring-context-3.2.3.RELEASE.jar
spring-core-3.2.3.RELEASE.jar
spring-expression-3.2.3.RELEASE.jar
spring-web-3.2.3.RELEASE.jar
spring-webmvc-3.2.3.RELEASE.jar
```

## 2. 한글인코딩방식

(1) POST 방식

web.xml에 다음 필터를 추가한다

```
<filter>
  <filter-name>CharacterEncodingFilter</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
    <!--<param-value>EUC-KR</param-value>-->
  </init-param>
```

```
<init-param>
  <param-name>forceEncoding</param-name>
  <param-value>true</param-value>
</init-param>
</filter>
<filter-mapping>
  <filter-name>CharacterEncodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

## (2) GET 방식

GET방식의 경우 주소창으로 파라미터값이 전달되는 경우라 코드상에 해결하기 보다는 WAS 설정을 통하여 해결하는것이 편리하다.

WAS가 톰캣이라면 다음과 같이 설정하면된다.

Tomcat 설정 파일인 server.xml 수정

Connector element의 attribute에 URIEncoding="UTF-8" 을 아래와 같이 추가해 주어야 한다.

```
<Connector connectionTimeout="20000" port="8989" protocol="HTTP/1.1" redirectPort="8443"
URIEncoding="UTF-8"/>
```



## 예제1. Spring MVC

### Dynamic Web Project : step14

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  
**Class File** : src/main/java/board.controller/LoginController.java  
src/main/java/board.controller/BoardListController.java  
src/main/java/board.controller/BoardViewController.java  
src/main/java/board.controller/BoardWriteController.java  
src/main/java/board.controller/BoardDeleteController

Folder : member  
JSP File : src/main/webapp/member/loginForm.jsp  
Folder : board  
JSP File : src/main/webapp/board/boardWriteForm.jsp  
src/main/webapp/board/boardDelete.jsp  
src/main/webapp/board/boardWrite.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

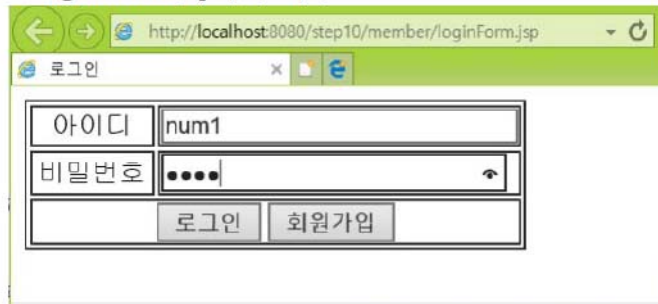
**XML File** : src/main/webapp/WEB-INF/web.xml  
src/main/webapp/WEB-INF/mvc-config.xml

#### <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
8. 우선 기존의 동작이 되는지 확인한다.
9. web.xml 파일 수정
10. 스프링 설정파일(XML 파일) 만들기
11. web.xml 파일에 한글 설정하기
12. 기존 클래스 파일 삭제하기
13. 기존 Controller 클래스들의 Controller 인터페이스를 spring 것으로 교체한다.
14. HandlerMapping 등록

## <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](#)

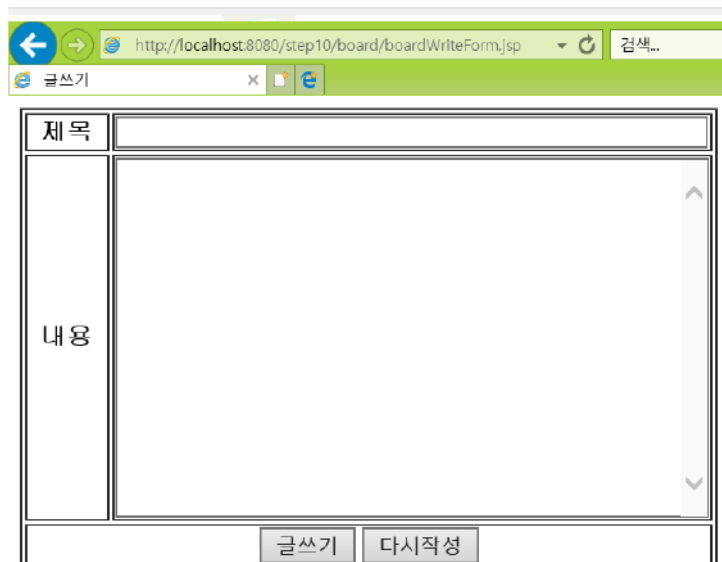


내일은

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

웃으리...

목록 글수정 글삭제



제목

내용

글쓰기 다시작성

## 17장. spring\_step15\_Spring MVC (2)

- Spring MVC도 스프링 설정 파일에 HandlerMapping, Controller, ViewResolver를 어노테이션으로 설정할 수 있다.

### 예제1. Spring MVC에 어노테이션 적용하기 (1)

Dynamic Web Project : step15

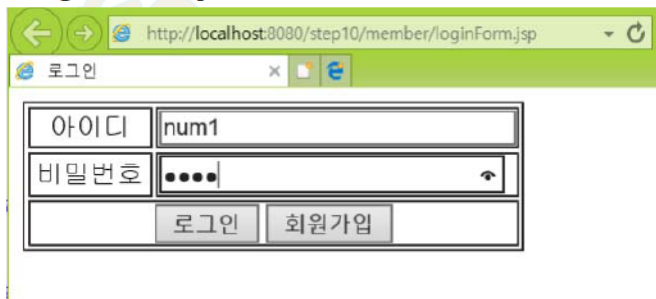
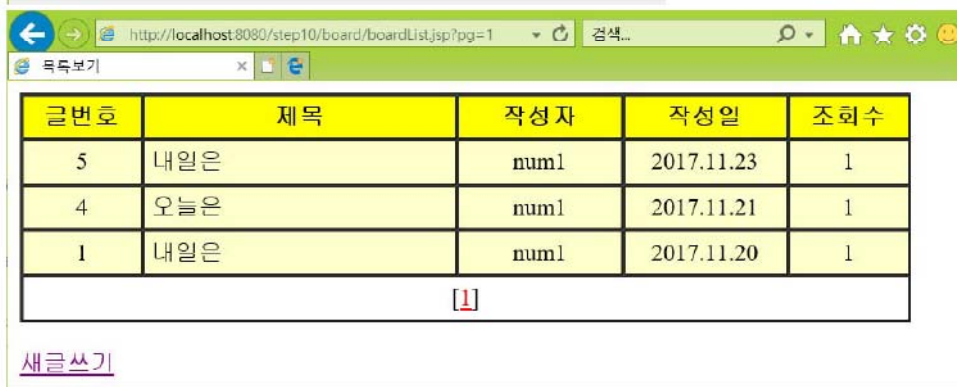
Package	: member.dao
Class File	: src/main/java/member.dao/MemberDAO.java
Package	: board.bean
Class File	: src/main/java/board.bean/BoardDTO.java
Package	: board.dao
Class File	: src/main/java/board.dao/BoardDAO.java
Package	: board.controller
Class File	: src/main/java/board.controller/LoginController.java src/main/java/board.controller/BoardListController.java src/main/java/board.controller/BoardViewController.java src/main/java/board.controller/BoardWriteController.java src/main/java/board.controller/BoardDeleteController
Folder	: member
JSP File	: src/main/webapp/member/loginForm.jsp
Folder	: board
JSP File	: src/main/webapp/board/boardWriteForm.jsp src/main/webapp/board/boardDelete.jsp src/main/webapp/board/boardWrite.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
XML File	: src/main/webapp/WEB-INF/web.xml src/main/webapp/WEB-INF/mvc-config.xml

## <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. web.xml 파일 수정
7. web.xml 파일에 한글 설정하기
8. Class 파일 추가하기
9. jsp 파일 추가하기
10. 우선 기존의 동작이 되는지 확인한다.
11. mvc-config.xml에 어노테이션 설정
12. Controller 클래스에 @Controller 어노테이션 추가한다.
13. @RequestMapping을 사용하려고 하는 메소드에 추가한다.

## <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](#)

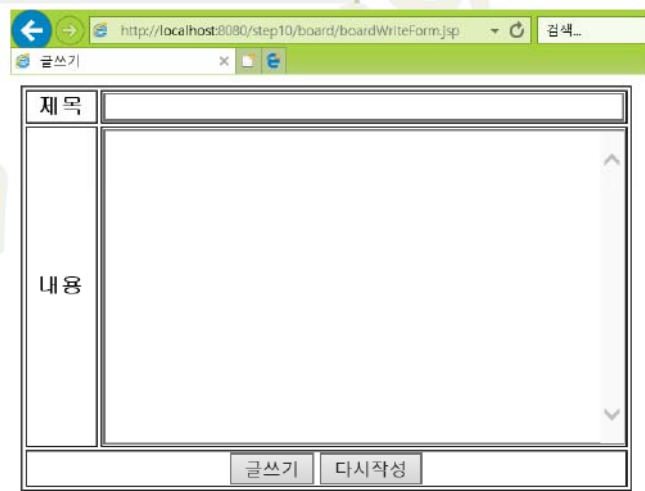


내일은

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

웃으리...

목록 글수정 글삭제



제목	
내용	

글쓰기 다시작성



## 18장. spring\_step16\_Spring MVC (3)

### \* Command 객체

- Command 객체는 @RequestMapping가 붙은 메소드의 매개변수에 전달되어지는 객체를 관리한다.

### 예제1. Spring MVC에 어노테이션 적용하기 (2)

#### Dynamic Web Project : step16

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
Class File	: src/main/java/board.controller/LoginController.java src/main/java/board.controller/BoardController.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/boardDelete.jsp src/main/webapp/board/boardWrite.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
XML File	: src/main/webapp/WEB-INF/web.xml src/main/webapp/WEB-INF/mvc-config.xml



## <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. web.xml 파일 수정
7. web.xml 파일에 한글 설정하기
8. Class 파일 추가하기
9. jsp 파일 추가하기
10. 우선 기존의 동작이 되는지 확인한다.
11. mvc-config.xml에 어노테이션 설정
12. Controller 클래스에 @Controller 어노테이션 추가한다.
13. @RequestMapping을 사용하려고 하는 메소드에 추가한다.
14. BoardController 클래스 추가

## <실행 결과>

- loginForm.jsp에서 시작

로그인

아이디: num1

비밀번호: ....

로그인 회원가입

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

글쓰기

제목	
내용	

글쓰기 다시작성

## 19장. spring\_step17\_Spring MVC (4)

### \* Service

DAO와 Service는 그 역할이 다르다.

DAO는 단일 데이터 접근/갱신만 처리한다.

Service는 여러 DAO를 호출하여 여러번의 데이터 접근/갱신을 하며 그렇게 읽은 데이터에 대한 비즈니스 로직을 수행하고, 그것을 하나의(혹은 여러개의) 트랜잭션으로 묶는다.

즉, Service가 트랜잭션 단위이다.

위와 같이 DAO와 Service가 완전히 동일해지는 경우도 발생한다.

하지만 그것은 해당 비즈니스 로직이 "단일 DB 접근"으로 끝나기 때문에 발생하는 것이다.

- Controller에서 Service를 통해 DAO를 사용하는 것은, 유지보수의 편의성 때문이다.

즉, Controller를 건드리지 않고서, DAO를 쉽게 교체하기 위해서이다.

- AOP 적용을 일반적으로 Service 구현 클래스의 메소드에 설정한다.

### \* 비즈니스 로직의 설정파일 사용

- web.xml 파일의 설정을 통해 비즈니스 로직의 클래스를 제어하는 application-config.xml을 사용할 수 있다.

- 설정

```
<context-param>
```

```
<param-name>contextConfigLocation</param-name>
```

```
<param-value>classpath:spring/application-config.xml</param-value>
```

```
</context-param>
```

```
<listener>
```

```
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
```

```
</listener>
```

## 예제1. Service 적용하기

### Dynamic Web Project : step17

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
Interface File	: src/main/java/board.controller/BoardService.java
Class File	: src/main/java/board.controller/LoginController.java
	src/main/java/board.controller/BoardController.java
	src/main/java/board.controller/BoardServiceImpl.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/boardDelete.jsp
	src/main/webapp/board/boardWrite.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
XML File	: src/main/resources/spring/application-config.xml
XML File	: src/main/webapp/WEB-INF/web.xml
	src/main/webapp/WEB-INF/mvc-config.xml

#### <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. web.xml 파일 수정
7. web.xml 파일에 한글 설정하기
8. Class 파일 추가하기
9. jsp 파일 추가하기
10. 우선 기존의 동작이 되는지 확인한다.
11. mvc-config.xml에 어노테이션 설정
12. BoardService 인터페이스를 추가한다.
13. BoardServiceImpl 클래스를 추가한다.
14. application-config.xml에 어노테이션 설정
15. BoardController, BoardDAO 클래스를 수정한다.

## <실행 결과>

- loginForm.jsp에서 시작

로그인

아이디 num1

비밀번호 .....

로그인 회원가입

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

## 20장. spring\_step18\_Spring MVC - MyBatis

### 1. MyBatis

- 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 정 이해해야 한다.
- MyBatis에서는 SqlSession을 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. MyBatis 적용하기

### Dynamic Web Project : step18

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  
Interface File : src/main/java/board.controller/BoardService.java  
Class File : src/main/java/board.controller/LoginController.java  
src/main/java/board.controller/BoardController.java  
src/main/java/board.controller/BoardServiceImpl.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/boardDelete.jsp  
src/main/webapp/board/boardWrite.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  
  
**Properties File : src/main/resources/spring/db.properties**  
**XML File : src/main/resources/spring/application-config.xml**  
**XML File : src/main/webapp/WEB-INF/web.xml**  
src/main/webapp/WEB-INF/mvc-config.xml  
**XML File : src/main/resources/mybatis/board-mapping.xml**  
src/main/resources/mybatis/mybatis-config.xml

## <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. web.xml 파일 수정
7. web.xml 파일에 한글 설정하기
8. Class 파일 추가하기
9. jsp 파일 추가하기
10. 우선 기존의 동작이 되는지 확인한다.
11. mvc-config.xml에 어노테이션 설정
12. application-config.xml에 어노테이션 설정
13. MyBatis SQL Mapper XML 파일 작성
14. MyBatis 환경설정 파일 작성
15. BoardDAO 클래스를 수정한다.

## <실행 결과>

- loginForm.jsp에서 시작

아이디: num1  
비밀번호: .....  
로그인 회원가입

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

글쓰기

제목	
내용	

글쓰기 다시작성

## 21장. spring\_step19\_fileupload 1

### \* CommonsMultipartResolver

- 스프링 설정 파일에 CommonsMultipartResolver를 <bean> 등록을 해야한다.

```
<bean id = "multipartResolver"
      class = "org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <property name="maxUploadSize" value = "5242880"/>
    <property name="defaultEncoding" value = "UTF-8"/>
</bean>
```

- id는 반드시 "multipartResolver"를 써야 한다.  
DispatcherServlet에 id가 미리 등록되어 있기 때문이다.

### \* MultipartFile

- 스프링 컨테이너에 의해 생성된다.
- 스프링 컨테이너는 "multipartResolver"로 등록된 CommonsMultipartResolver를 통해서 MultipartFile 객체를 생성한다.
- 클라이언트에서 전달된 파일에 대한 모든 정보가 저장되어 있다.

## 예제1. file upload

Project Name : step19  
package Name : imageboard.bean, imageboard.controller  
class Name : src/main/java/imageboard.bean/ImageboardDTO.java  
src/main/java/imageboard.controller/ImageboardController.java  
Folder Name : storage (파일저장 폴더)  
webapp/storage  
Folder Name : imageboard  
JSP File : webapp/imageboard/imageboardWriteForm.jsp  
webapp/imageboard/imageboardWrite.jsp  
XML File : webapp/WEB-INF/mvc-config.xml  
webapp/WEB-INF/web.xml

### <작업 순서>

1. 프로젝트 만들기
2. JRE System Library 버전 변경하기
3. pom.xml 파일 수정하기
4. pom.xml 파일에 추가하기
5. web.xml 파일에 한글 설정하기
6. mvc-donfig.xml에 설정하기
7. storage 폴더 만들기
8. Class 파일 추가하기
9. jsp 파일 추가하기

### <실행 결과>

상품 코드	img_notebook
상품명	노트북
단가	1200000
개수	5
내용	성능 좋은 노트북

D:\노트북.jpg    찾아보기...

이미지등록    다시작성

