

- Mybatis

SqlSession 객체 – mybatis 제공 api(SqlSession)--> spring 제공 api(SqlSessionTemplate)

```
List<resultType 객체> <--- session.selectList( "xml 태그id" )
resultType 객체 <--- session.selectOne("xml 태그id")
session.selectList( "xml 태그id" , 객체, 배열, 1개값 )
session.selectOne("xml 태그id", 객체, 배열, 1개값 )

=====
int <--- session.insert( "xml 태그id" )
int <--- session.insert( "xml 태그id", 객체, 배열, 1개값 )
int <--- session.update( "xml 태그id" )
int <--- session.update( "xml 태그id", 객체, 배열, 1개값 )
int <--- session.delete( "xml 태그id" )
int <--- session.delete( "xml 태그id", 객체, 배열, 1개값 )
session = factory.getSession(true);
=====session.commit() =====
```

24장 spring() main + mybatis 연동하기

<p>String main 새로 작성 클래스 ApplicationContext factory = new ClassPathXmlApplicationContext (spring bean configuration xml);</p>	<p>db-config.xml(이동) sql-mapping.xml(그대로) spring bean configuration xml 새로 작성 1>spring제공 datasource 2>spring mybatis 2개xml 파일명.. 3> annotation인식패키지 설정</p>	<p>@Service MemberServiceImpl @Autowired @Repository MemberDAO @Component MemberDTO : SELECT <--> INSERT,UPDATE</p>
<p>pom.xml</p>	<p>mysql-connector-javaxx.jar mybatisxxx.jar 라이브러리 mybatis-springxxx.jar spring-jdbcxxx.jar</p>	

springmain

MemberService

@Service("service")
MemberServiceImpl
@Autowired
MemberDAO

sql-mapping.xml(그대로)

mybatis-config.xml:(TYPEALIAS
제외하고 삭제)

@Repository
MemberDAO
@Autowired
SqlSession



MemberDTO

spring-mybatis.xml :

- 1> SPRING DATASOURCE
- 2> SQLSESSIONFACTORY(2개의 마이바티스 XML파일)
- 3> sqlSession 생성
- 4> annotation 인식 패키지 설정

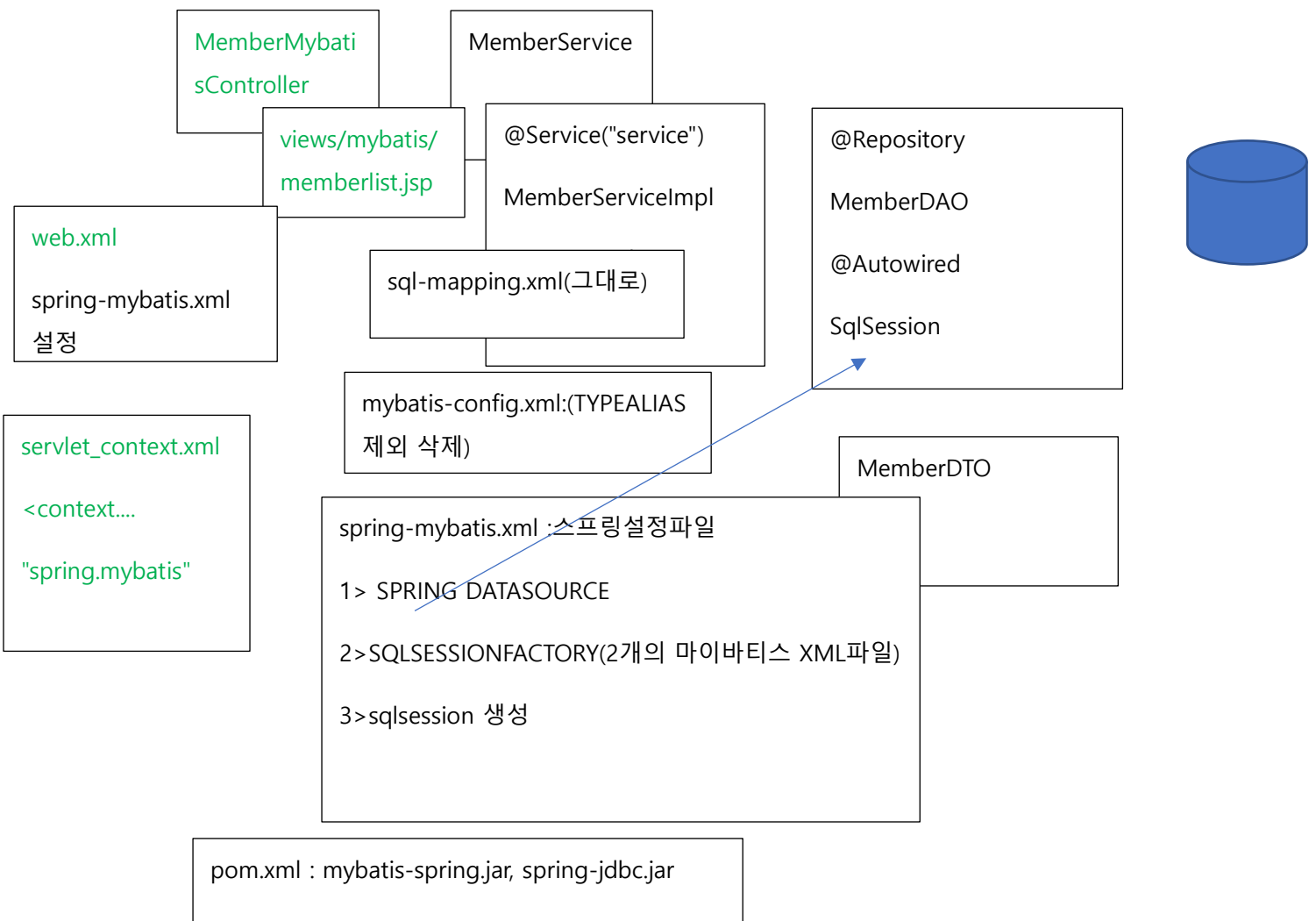
pom.xml :

mysql-connector-java.jar , mybatis.jar

mybatis-spring.jar , spring-jdbc.jar

spring mvc + mybatis

<http://localhost:8080/test/membermybatislist>



- 스프링 mvc와 mybatis 활용한 게시판
오후 소회의실 4개 - 1시간 주제 논의

<http://localhost:880/test/boardstart>

나의 게시판 시작화면입니다.

메뉴

1. 게시물관리보기(로그인하지 않아도 됩니다)

---> 화면 내부

게시물 검색폼

게시물관리 5개 (1페이지당)

테이블태그 - 번호 제목링크 작성자 조회수

제목클릭 - 상세게시물보기 - 번호 제목 내용 작성자 조회수 작성시간 - 수정 삭제 (게시글 암호)

수정클릭 - 수정폼 - 수정반영

삭제클릭 - 글번호 삭제반영

페이지번호

2. 글쓰러가기(로그인하셔야 가능합니다)

글쓰기폼 - 작성자(로그인한 사용자 아이디 미리 입력)

제목 내용 글암호 입력(번호자동증가, 조회수(0), db now())

3. 로그인하기 HttpSession

4. 로그아웃하기

테이블구성

member 테이블

board 테이블

번호	seq int auto increment primary key
제목	title varchar(100) not null
내용	contents varchar(4000)
작성자	writer varchar(30) member테이블의 id 컬럼 존재값 사용가능
암호	pw int
조회수	viewcount int
작성시간	writingtime datetime now()

/	시작화면
/boardwrite : get	글쓰기폼
/boardwrite : post	글쓰기처리완료
/boardlogin : get /post	로그인하기 session.set...
/boardlogout	로그아웃하기 session.remove
/boardlist	리스트보러가기
/boarddetail	선택 게시물 모든 컬럼 정보 조회
BoardDTO클래스	BOARD 테이블 1개 레코드 객체

BoardController-메소드들 각 테이블당 crud

MemberController-메소드들 로그인 로그아웃

model 없다

board/start.jsp

/boardwrite : get 컨트롤러 메소드 String 메소드이름 ()	board/writingform.jsp 뷰 action=boardwrite method=post 제목 내용 글암호-숫자 작성자-로그인한아이디미리입력 입력
/boardwrite : post ModelAndView 메소드이름 (BoardDTO dto)	dto 동일명 변수에 전달받은 폼데이터들 동일명 파라미터값들 저장 db board 테이블 insert BoardService:xxxxx BoardSeviceImpl BoardDAO board-mapping.xml 실행결과 insert 행 갯수 모델로 저장 board/writingprocess.jsp 뷰 게시물 1개 저장했습니다.

<p>/boardlist ModelAndView 이름(int page) @RequestParam(value="page", required=false, defaultVaue="1") 추가</p> <p>SERIVE, DAO , BOARD-MAPPING. XML 추가 메소드</p> <p>SELECT COUNT(*) FROM BOARD</p> <p>SELECT * FROM BOARD ORDER BY WRITINGTIME DESC LIMIT ?, 3</p>	<p>board/list.jsp 구성 1. page 번호 입력 페이지 게시물 3개씩 출력 테이블 태그 구성</p> <p>페이지번호 클릭 a 태그 제공 (게시물저장갯수에 따라) <a href="boardlist?page=</p>
--	---

<p>/boarddetail ModelAndView 이름(int seq)</p> <p>1> 해당게시물 조회수 +1 sql 실행 update board set viewcount = viewcount + 1 where seq=#{seq}</p> <p>2> 조회 리턴 select * from board where seq=#{seq} 실행</p> <p>SERVICE(1개), DAO(2개) , BOARD-MAPPING. XML(2개) 추가 메소드</p>	<p>board/detail.jsp 번호: 제목: 내용: 작성자: 조회수: 작성시간:</p> <p>수정버튼 삭제버튼</p>
--	--

흐름도

controller 메소드 매핑 - jsp 화면구성

컨트롤러 갯수 이름, jsp

패키지명.클래스명. jsp파일명규칙, js파일, jquery

