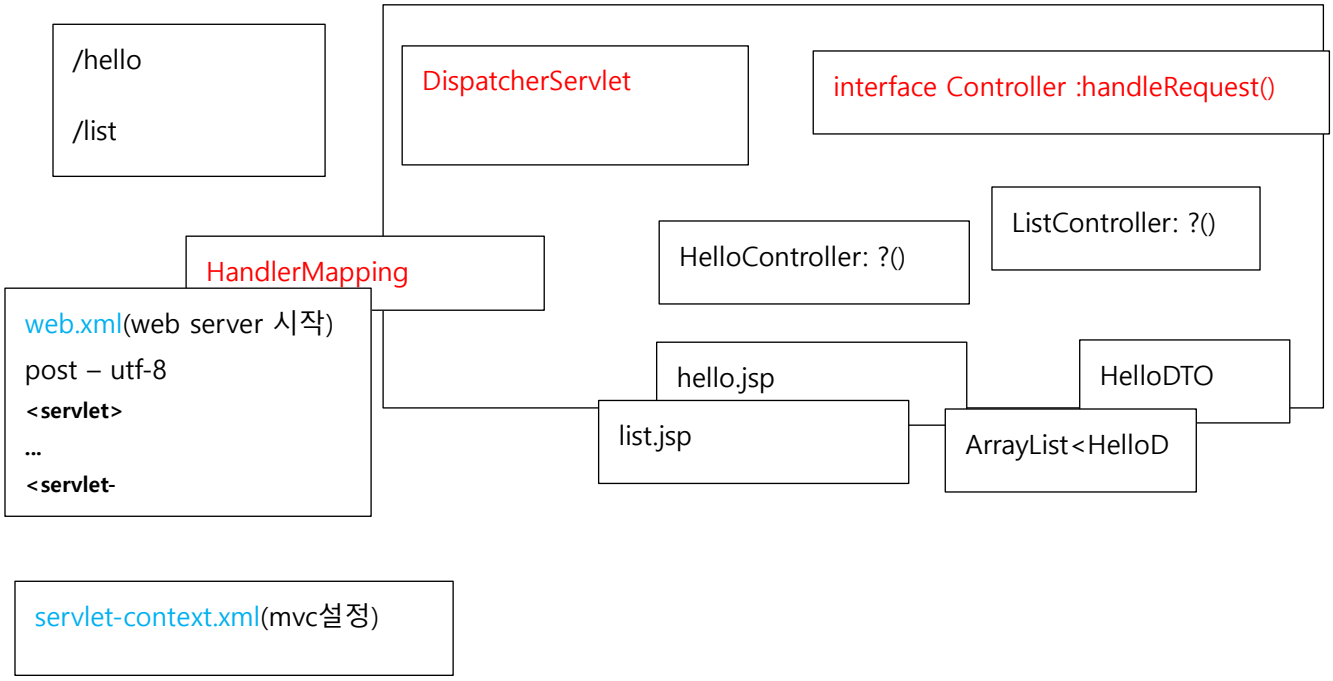
spring mvc – 웹서버 개발 패턴

요청 – 분석 – 처리 controller 선정 =FrontController



annotation

xml 21장	annotation 26장
class A implements Controller{ ModelAndView handleRequest (request, response){ } class B implements Controller{ ModelAndView handleRequest (request, response){ } <beans:bean id="a" class="....A" > <beans:bean id="b" class="....B" > < beans:bean SimpleUrlHandlerMapping	@Controller class A { @RequestMapping("/a") @GetMapping @PostMapping LoginDTO m1(@ModelAttribute("dto") XXDTO){ } @RequestMapping("/b") String m2(@RequestParam("s2") String s){} ..... } xml- <context:component-scan

< ...key="/a"> a < ...key="/b"> b	base-package="패키지명" />

- @Controller @RequestMapping 메소드 리턴타입

ModelAndView	모델o, 뷰o
String	모델x, 뷰o.뷰이름 지정
return "redirect:/loginform"	/loginform url 매핑 메소드 호출
void	모델x, 뷰o. 자동 mapping url 동일명 jsp
그밖의 기타 다른 리턴타입	객체리턴, 뷰 별도 존재하기보다는 현재 뷰의 특정 태그 내부 포함 내용 전달 =ajax

- @Controller @RequestMapping 메소드 매개변수

x	가능
servlet api	HttpServletRequest HttpServletResponse 가능, 지양-응답(jsp)
String , String[]	form <input name="id" String id
form->DTO->db insert	DTO 객체내부변수이름 = form <input name=" 이름"
Map	("id", id)

- spirng mvc annotation

@Component @Repository @Service @Autowired @Qualifier

@Controller	컨트롤러 클래스 위
@RequestMapping("url")	메소드 위
@RequestParam("요청파라미터명")	메소드 매개변수 앞 (@RequstParam("id2") String id) html --> 컨트롤러 전달
@ModelAttribute("모델명")	메소드 매개변수 앞 (@ModelAttribute("dto") MemberDTO) html ---> 컨트롤러 --> jsp 뷰 전달
@RequestMapping(value="url", method=RequestMethod.GET|POST) @GetMapping("") @PostMapping("")	 <form action="" mehtod="Post"

- spring mvc 폴더구조

src/main/java	패키지명.*.java dto, dao, service, controller, 서블릿.... spring mvc <context:component-scan base-pacKage=" 패키지명 "
src/main/webapp/resources	html css js jpg mp3
src/main/webapp/WEB-INF/views	jsp
src/main/webapp/WEB-INF/spring	appServlet₩servlet-context.xml ==> spring mvc 1> resources 폴더 html관련내용 저장 2> view 저장폴더,확장자 설정 3> annotation 인식 패키지 지정 4>
src/main/webapp/WEB-INF/web.xml	프로젝트 tomcat 서버 설정 url->DispatcherServlet
프로젝트루트/pom.xml	spring 라이브러리 버전, 다운로드 관리 aop-추가 다운로드 <dependency> 추가

23장 마이바티스 프레임워크 사용하기

1> jdbc 개선방법- sql 별도 분리 xml 파일 따라 저장

2> 스프링 아니다

3> 스프링은 di, ioc, aop, mvc, 다른 프레임워크 연동 여러 기능 제공 통합 프레임워크

4> mybatis db연동 프레임워크(MemberDAO 수정+MYBATIS 설정 XML)

5> spring mvc + spring 의 다른 프레임워크연동 + mybatis

2> Mybatis

jdbc ---> mybatis 변경

SQL 별도 분리 작성 – SQL MAPPER FRAMEWORK

IBATIS(회사) --> Mybatis(회사)

org.ibatis.xxxx

- mybatis 시작

1>
mysql jdbc driver.jar-->pom.xml
mybatis.jar -->pom.xml

2> sql xml파일—sql mapping 파일 작성
3> db config xml 파일 – db 연결정보--> mybatis DataSource 이용 설정)
4> Main에서
Service / DAO / DTO

sql xml 파일
crud
<select id="" resultType="" [parameterType=""]
<insert id="" [parameterType=""]
<update id=""[parameterType=""]
<delete id="" [parameterType=""]

<foreach items="array" separator="," var="id" >
#{id}

.....===> sql
select * from emp order by id limit 10, 3;
select * from emp where id in (100, 200, 300, 400) ;

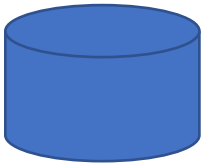
자바코드
session.selectList("test", new int[]{100, 200, 300, 400});

<select id="test" resultType="EmpDTO" >
select * from emp where id in
<foreach items="array" begin="(" separator="," end=")" var="id" >
#{id}

</select>

workbench tool

INSERT/DELETE/UPDATE(DML)- 바로 반영



JDBC – JDK

INSERT/DELETE/UPDATE(DML)- 바로 반영

MYBATIS

INSERT/DELETE/UPDATE(DML)- 임시 버퍼 저장(1)

MAIN종료- 취소

1. COMMIT – DB반영

2. ROLLBACK – 취소

sql-mapping.xml
db-config.xml -> (db연결정보+sql매핑파일명)

SqlSession 객체

List<resultyType 객체> <--- session.selectList("xml 태그id")
resultyType 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() ==========

create table member
(id varchar(30) not null primary key,

alter table member add constraint primary key(id);
==>
primary key --> id값은 다른 레코드의 값 중복x

나중에 primary key 제약조건 추가시 중복 id값 없어야만 가능하다

mybatis member 테이블 crud
<select
<insert
<update
<delete
<foreach
<choose
 <when test="T/F">
   true 실행구문
 </when>
<when test="T/F">
   true 실행구문
 </when>

</choose>

java
MemberDTO dto
dto.setName("")

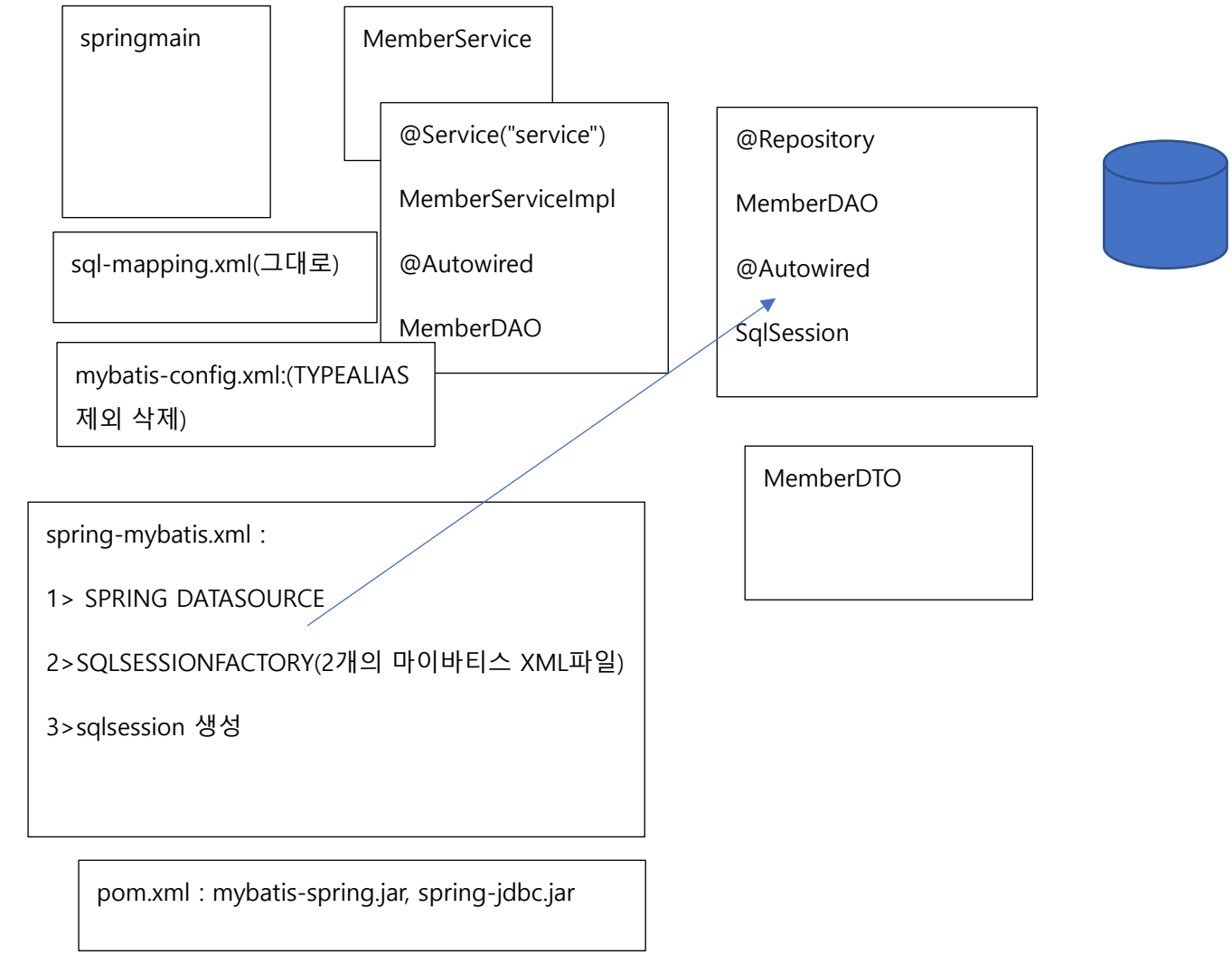
xml
update member
<choose>
<when test="name != null">
set name=#{name}
</when>
<when test="name != null and email!= null">
set name=#{name}, email =#{email}
</when>

where id=#{id}

24장 spring(주) main + mybatis(부속) 연동하기

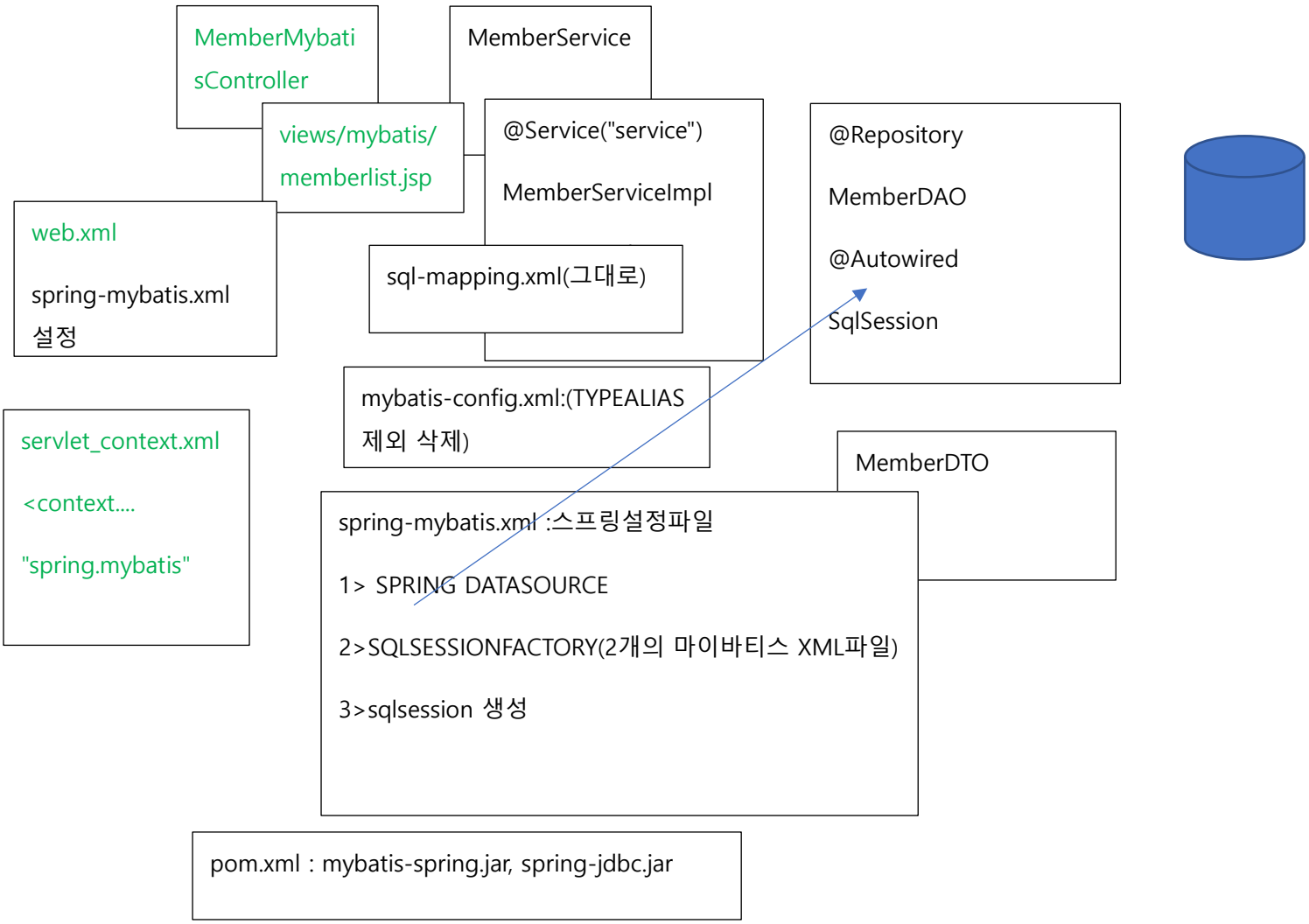
<table>
<tr><td>String main 새로 작성 클래스
ApplicationContext
factory
= new
ClassPathXmlApplicationContext
(spring bean configration xml
);</td><td>db-config.xml(일부설정)
sql-mapping.xml(그대로)
spring bean configration xml
새로 작성

1>spring제공 datasource
2>spring mybatis 2개xml 파일명..
3> annotation인식패키지 설정</td><td>@Service
MemberServiceImpl
@Autowired
@Repository
MemberDAO
@Component
MemberDTO : SELECT <-->
INSERT,UPDATE</td></tr>
<tr><td>pom.xml</td><td colspan="2">mysql-connector-javaxx.jar
mybatisxxxx.jar 라이브러리
spring mybatis 연동 라이브러리
mybatis-springxxx.jar
spring-jdbcxxxx.jar</td></tr>
</table>


springmain
MemberService
@Service("service")
MemberServiceImpl
@Autowired
MemberDAO
@Repository
MemberDAO
@Autowired
SqlSession
sql-mapping.xml(그대로)
mybatis-config.xml:(TYPEALIAS 제외 삭제)
MemberDTO
spring-mybatis.xml :
1> SPRING DATASOURCE
2>SQLSESSIONFACTORY(2개의 마이바티스 XML파일)
3>sqlsession 생성
pom.xml : mybatis-spring.jar, spring-jdbc.jar

spring mvc + mybatis

http://localhost:8080/test/membermybatislist



MemberMybatisController

1> 메소드

url 매핑 - /membermybatissearchlist

item/searchword 파라미터로 전송

조건 맞는 데이터만 검색

memberlist.jsp 결과 리턴

MemberService / MemberServiceImpl/MemberDAO

2>

sql-mapping.xml

select * from member where ????? like '%' || #{searchword} || '%'

<choose> 880p 참고

<when>