

1. spring mvc + 비지니스 로직(처리)- SERVICE + DAO + DTO

java application + 비지니스 로직(처리)

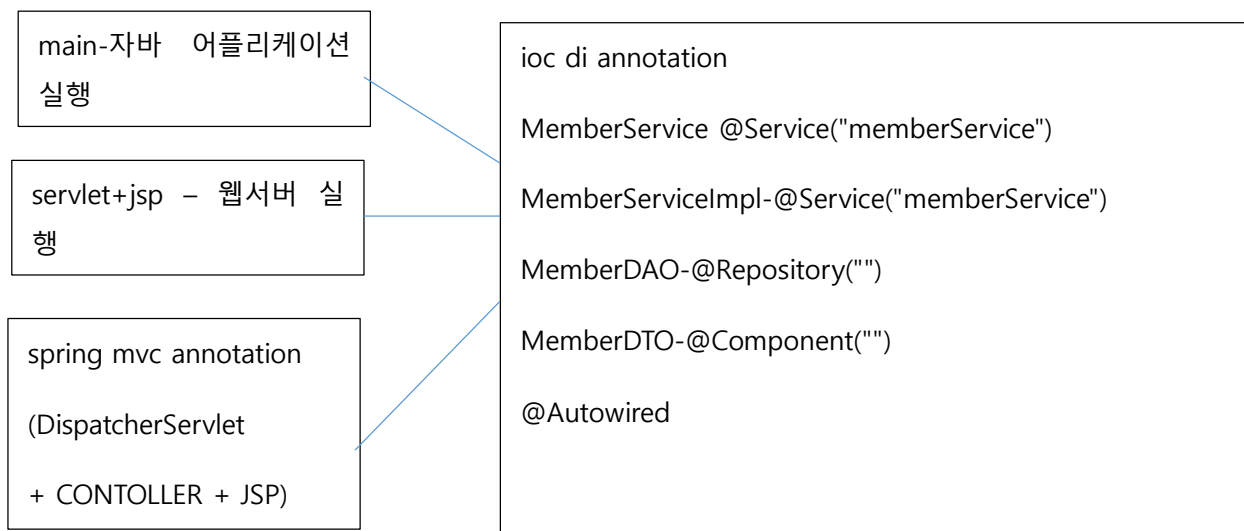
servlet + jsp + 비지니스 로직(처리)

2. MYBATIS "DB" 프레임워크-SQL 만 정의 XML

- STS4 TOOL SPRING MVC 프로젝트의 XML파일들

pom.xml	프로젝트 root위치 스프링 +MYBATIS 필요 라이브러리 다운로드 설정태그 추가-MAVEN 툴
web.xml	src/main/webapp/WEB-INF 위치 웹서버 설정 "/" --> DispatcherServlet 전송(servlet-context.xml설정 같이 읽어라) "/" -post - utf-8 인코딩 처리 --> DispatcherServlet + 스프링설정파일 추가 알려주는 파일 member.xml 파일 추가
servlet-context.xml	src/main/webapp/WEB-INF/spring/appServlet 위치 spring mvc BEAN 설정 파일
스프링설정파일 추가	경로 / *.xml src/main/java/annotation/service/member.xml ==> classpath: annotation/service/member.xml
	pom.xml web.xml servlet-context.xml mybatis-config.xml(db연결정보) sql-mapping.xml(sql 저장- 결과/파라미터)

- spring mvc와 비지니스 로직 처리 연동



```
@Controller
class A {
    @Autowired
    MemberService service;

    @RequestMapping("/uri")
    public void m(){
        uri.jsp 뷰
```

- spring mvc view에서 이미지, js, css 파일 사용

*.java(컨트롤러 service dto dao class)	src/main/java/패키지명.xxx
jsp	src/main/webapp/WEB-INF/views/*.jsp
html http://ip:port/mvc/resources/*.html css js gif jpg	src/main/webapp/resources/*.html

- db

spring mvc + mybatis

자바 jdbc jdbc 기능	java.sql.*
spring jdbc 기능	22장
mybatis 프레임워크	23, 24장 db 연동 기능만. spring mvc / di / 다른프레임워크통합 - mybatis연동
hibernate 프레임워크	
	25 db 트랜잭션 기능 mysql 트랜잭션처리 자동

mybatis framework

자바 작성. 대표적 특징은 SQL문을 분리하는 SQL Mapper 프레임워크.

```
<select id="s" resultType="MemberDTO" parameterType="String">
select * from member_table where id=#{a}
</select>
```

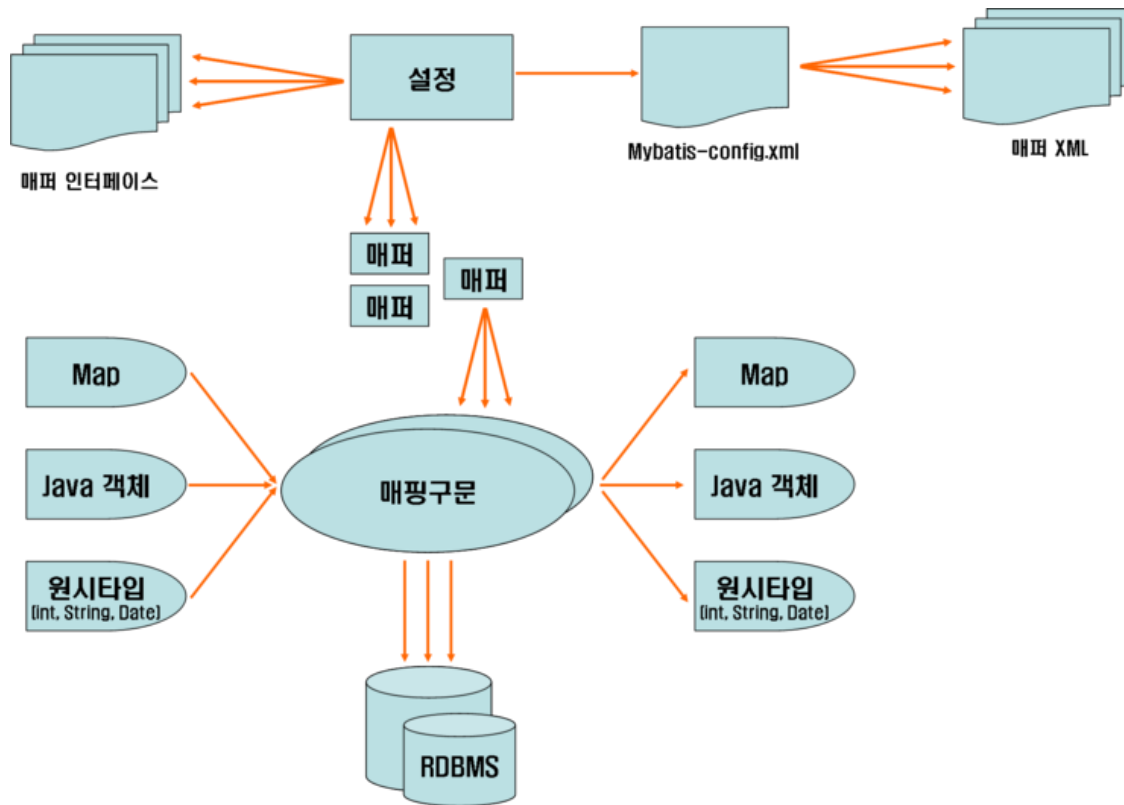
- MYBATIS 환경설정

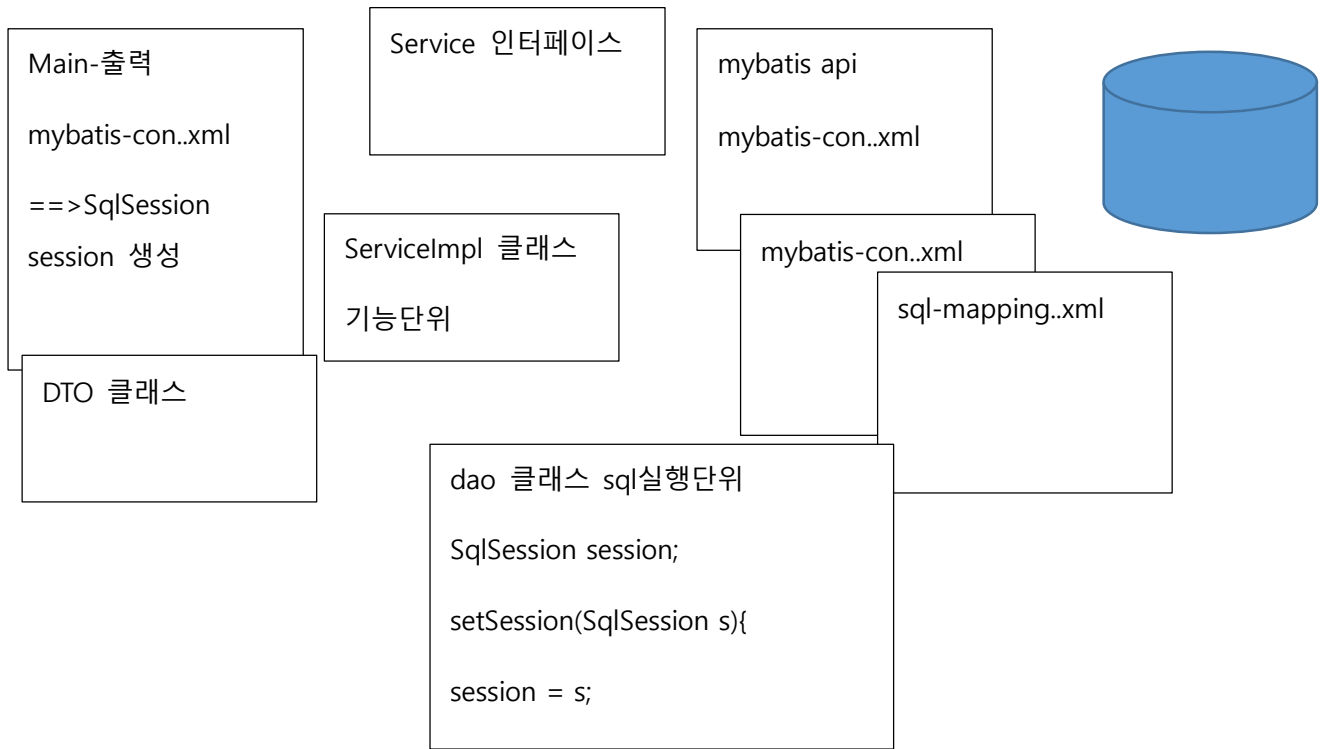
1> SPRING MVC + MAVEN TOOL 관리

2> pom.xml <dependency – 1쌍 1개.jar 다운로드 설정

MYBATIS 라이브러리 태그 1개

spring+mybatis 연동 라이브러리 태그 2개
mysql jdbc driver 태그 1개





24장 spring mvc와 mybatis 연동

1> "DB"-SQL 분리 XML 정의

2> BEAN 생성 - 관리 (SINGLETON) + @Component(객체생성처리) + mvc + ioc di + 다른 프레임워크 연동

spring main 메소드 + mybatis--> sts4 콘솔

mybatis연동 spring bean 설정 파일.xml	config.xml	sql-mapping.xml(그대로)
<p>....</p> <p>2. datasource정보 설정</p> <p>3. sql매핑파일명 설정</p> <p>4. SqlSession 객체 생성 <bean</p> <p>5.</p> <p><context:component-scan</p>	<p>...</p> <p><configurations></p> <p>1. spring.mybatis.MemberDTO</p> <p>==> "memberdto" 이름 설정</p>	<p>...</p> <p><mappers></p> <p><select</p> <p><insert</p> <p><update</p> <p><delete</p> <p><forEach</p>

1> mybatis 패키지 복사 -spring.mybatis 로 이름 변경

2> spring-mybatis-mysql.xml(스프링 빈 설정 파일) 생성

3> datasource

config파일

mapping파일

sqlTemplate생성

component scan

내용 설정

4> config파일--> typeAliases 제외하고 나머지 태그 삭제

5> mapping파일--> 그대로

6>

```
class MemberMain
ApplicationContext factory = new ClassPathXmlApplicationContext
("springmybatis/mysql/member/spring-mybatis.xml");
MemberService service = (MemberService)factory.getBean("mybatiservice");
.....
```

```
@Service("mybatiservice")
class MemberServiceImpl
@Autowired
@Qualifier("mybatisdao")
MemberDAO dao;
```

```
@Repository("mybatisdao")
class MemberDAO
@Autowired
SqlSession session;
```

7> 스프링 mvc --> 오류발생시 url이랑 @ 이름들 겹치는 부분 찾아서 해결

web.xml

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/spring/root-context.xml
    classpath:...../spring-mybatis.xml
  </param-value>
</context-param>
```

spring-mybatis.xml

```
....
<context:component-scan base-package="spring.mybatis" /> ==> 웹서버 시작시 반영( 스프링mvc 시작이전)
@Controller 제외
```

servlet-context.xml ==> 스프링mvc 요청 시작시 반영

```
<context:component-scan base-package="spring.mybatis" />
@Controller 스캔
```

MemberMybatisController

@Controller

```
public class MemberMybatisController {
    @Autowired
    @Qualifier("mybatiservice")
    MemberService service;
    @RequestMapping("/membermybatislist")
    public ModelAndView memberlist() {
        List<MemberDTO> list = service.memberlist(new int[] {5,6});
        ModelAndView mv = new ModelAndView();
        mv.addObject("memberlist", list);
        mv.setViewName("mybatis/memberlist");
        return mv;
    }
}
```

```

}

mybatis/memberlist.jsp
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
.....
<h1> 조회 결과입니다 </h1>
<c:forEach items="${memberlist}" var="dto" >
    <h3> 이름 ${dto.name } 주소 ${dto.address } </h3>
</c:forEach>

```

MemberMybatisController.java – 1개 테이블 crud기능 메소드들 매핑

/membermybatislist	memberlist() 호출 -모든 회원 정보 모델 생성	mybatis/memberlist.jsp
/membermybatispaginglist?page=x	memberlist(int page) 호출 -1 – id 알파벳 순서 정렬 select ... limit 0, 5; 2 – id 알파벳 순서 정렬 select ... limit 5, 5; 3 – id 알파벳 순서 정렬 select ... limit 10, 5; - 5명 회원 정보 모델 생성	mybatis/memberlist.jsp
/membermybatissearchlist? address=서울&address=대전 &address=제주 ==> input type=checkbox name=address value="서울">서울	memberlist(String address[]) 호출 해당 주소 사는 회원 정보 모델 생 성	mybatis/memberlist.jsp
/membermybatisinsert - get	insertform() 모델 없다 회원가입 6개값 입력 폼 생성 화면	mybatis/insertform.jsp
/membermybatisinsert - post	insertresult(MemberDTO dto) db insert 결과행 갯수 리턴 모델	mybatis/insertresult.jsp 모델 == 1 "정상 회원가입"
/membermybatisupdate?id=xxxx - get	updateform(String id) 모델 없다 회원정보 수정폼 생성 화면	mybatis/updateform.jsp 아이디:\${param.id} 이름 주소 항목 수정 화면
/membermybatisupdate - post	udpateresult(MemberDTO dto) 수정 결과행 갯수 리턴 모델	mybatis/updateresult.jsp 모델 == 1 "정상 회원정보 수정"
/membermybatisdelete?id=xxx	deleteresult(String id){ 1. delete sql 삭제.= 1개	mybatis/memberlist.jsp

	2. 모든 회원 정보 모델 1개	
--	----------------------	--

```

/memberstart
public ModelAndView memberstart(){

```

mybatis/memberstart.jsp

- 로그인
- 로그아웃
- 회원가입
- 회원조회

전체조회

페이지조회

주소조회

내정보조회-내정보수정

23,24장

26장 스프링 어노테이션 기능

27장 sts프로젝트+라이브러리 다운로드 관리 =>maven 내장 ==> pom.xml

```

28장
http 요청 파일 서버 전송 기능 없다
HttpServletRequest
sTRING<==getParameter("")
sTRING[]<==getParameterValues("")

```

시작라인 /membermybatislist?변수명=값& get post
요청헤더 getCookies()
post 요청바디 변수명=값
서버로 파일 전송 = 변수명=파일내용+길이+이름+...(여러개파트)

- 다른 api
- jsp 파일 업로드 api
- 스프링 파일 업로드 api

Multipartxxxx 클래스들 다수개

파일업로드

파일다운로드

1> pom.xml = maven 다운로드 요청

commons-io.jar

commons-fileupload.jar

2> src/main/webapp/web-inf/spring/appServlet/servlet-context.xml – C...MultipartResolver

client	server
<pre><form action="/a" method="post" enctype="multipart/form-data" > <input type="file" name="f"> <INPUT TYPE=TEXT <input type="submit" value="파일전송"> </form></pre>	<pre>@Controller class A @RequestMapping("/a") public void m(Multipart f){ f.getOriginalFileName() f.getLength() f.transfer(다른파일명 서버 저장) }</pre>

- 온라인지원서 사진파일 / 게시판 첨부파일 / 쇼핑몰관리자 상품사진 수시로 업데이트
- 네이버 ai

UploadController	<pre>@RequestMapping(value="fileupload", method=....GET) 메소드(){ 모델 없다 uploadform.jsp 뷰 }</pre> <pre>@RequestMapping(value="fileupload", method=....POST) 메소드(UploadDTO dto){ 파일을 서버 c:\upload 폴더 파일 저장 처리결과 모델 정의 uploadresult.jsp 뷰 }</pre>
UploadDTO	
uploadform.jsp	
uploadresult.jsp	

29장 스프링 ajax 호출

=====

30장 게시판 답변형+첨부파일

31장 쇼핑몰

==> Member

=====

32장 스프링 부트

1>스프링 legacy project-spring mvc project --> 설정 복잡

2> 스프링 부트 --> 설정 간소

3.백엔드 프로젝트 (4일)--> 스프링 legacy project-spring mvc project

네이버 ai 접목 + spring boot

react, vue, node

최종프로젝트 --> spring boot

1페이지 0,3

2페이지 3,3

3페이지 6,3

(page-1)*3

-파일 업로드

서버 이력서 / 사진 전송 / 상품 이미지 / 게시판 첨부파일 업로드

- Multipart

- 웹 클라이언트가 요청을 보낼 때, **http 프로토콜의 바디** 부분에 데이터를 여러 부분으로 나눠서 전송
- 보통 파일을 전송할 때 http 프로토콜의 바디 부분에 파일정보를 담아서 전송을 하는데, 파일을 한번에 여러개 전송을 하면 body 부분에 파일이 여러개의 부분으로 연결되어 전송
- 여러 부분으로 나뉘어서 전송되는 것을 **Multipart data** 라고 한다.
- javax.servlet.http.HttpServletRequest 는 파일 업로드를 지원하지 않는다.
- 사용자는 javax.servlet.http.HttpServletRequest 의 body 부분을 읽어들이는 입력스트림을 이용해서 Multipart 부분을 잘 나누어서 사용해야 하는데 직접 구현하지 않고 라이브러리를 사용한다.

=== file upload

1>html 태그 업로드 기능

<form action="..."

method="post"

enctype="multipart/form-data"

<input type="file" --> 파일선택열기창

서버 파일 전송받는 기능-spring api

2> pom.xml – 27 장

스프링 라이브러리 관리

file upload / ajax / mybatis 추가 라이브러리 설정

```
<!-- 파일 업로드 -->
```

```
<!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->
```

```
<dependency>
```

```
    <groupId>commons-fileupload</groupId>
```

```
    <artifactId>commons-fileupload</artifactId>
```

```
    <version>1.3.1</version>
```

```
</dependency>
```

```
<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
```

```
<dependency>
```

```
    <groupId>commons-io</groupId>
```

```
    <artifactId>commons-io</artifactId>
```

```
    <version>2.6</version>
```

```
</dependency>
```

```
3>servlet-context.xml
```

```
    <beans:bean id="multipartResolver"
```

```
        class="org.springframework.web.multipart.commons.CommonsMulti  
partResolver" />
```

```
4> <input type=file name="f" > ---> MultipartFile f 매개변수로 선언
```

```
5> MultipartFile 메소드
```

```
getOriginalFilename()-> 전송파일명
```

```
transfer();-> 서버내부 내용 저장
```

29장 restful 기능 mvc

rest

a.jsp

이미지

댓글 10개

더보기

<div>????? </div>

b.jsp

댓글 10개

http

1> 요청 - 응답

2> 기본적 요청 처리 시간 동안 클라이언트 대기한다. 화면 아무 변화없다

---> 동기화 통신 방식(synchronous)

3> 변경- 요청- 화면 - 응답 - 변경

---> 비동기화 통신 방식(asynchronous)

--> java script and

--> xml (비동기통신시 전달 데이터형태 초기-)

--> 자바 스크립트 객체{"a" : "test" , "b":100, "c":true, "d": {.....} }

--> json--> java script object notation

--> excel, json

a = { "b" : "값" , }

=> a.b

(http 이전 상태정보 유지할 수 없으니 추가 HttpSession 구현)

json

ajax

@ResponseBody

@Controller

```
class A{
```

```
@RequestMapping("/a")
```

```
String m(){
```

```
    return "a";-->view이름
```

```
}
```

```
@RequestMapping("/b")
```

```
@ResponseBody
```

```
String m2(){
```

```
    return "{ \"\" : \"\", \"\" : \"\" }";==> "b" 결과리턴
```

```
}
```

- ajax

-ajax

1>synchronous javascript and xml

2>서버와의 비동기적 통신 방식

3> 화면 렌더링은 미리 해놓고 서버와의 응답 결과로 화면 갱신

4. 서버와의 데이터는 JSON 형식으로 설정

1> pom.xml

ajax하려면 우리가 컨트롤러에서 리턴하는 자바 객체를 자동으로 json 으로 변환해주는 라이브러리 필요

```
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.9.3</version>
</dependency>
```

2> \$.ajax / @ResponseBody

클라이언트- 자바스크립트와 jquery	<pre>\$.ajax({ url: "<%=request.getContextPath()%>/ajax/login", data: { 'id':\$("#id").val(), 'pw':\$("#pw").val() }, type: 'post' , dataType: 'json' , success: function(server){ "{id":"xxx"}" var obj = JSON.parse(server); });</pre>
-----------------------------	--

서버-스프링	<pre> @RequestMapping(value="/ajax/login", method = RequestMethod.POST , produces ={"application/json;charset=utf-8" }) @ResponseBody public String loginresult(String id, String pw){ return ""; } </pre>
문자열을 json 객체 형변환	<pre> var obj = JSON.parse(server); </pre>
json을 문자열로 형변환	<pre> JSON.stringify(json객체) </pre>