스프링 기술

aop	mvc	spring jdbc	프레임워크연동
기본 필수 ioc di			

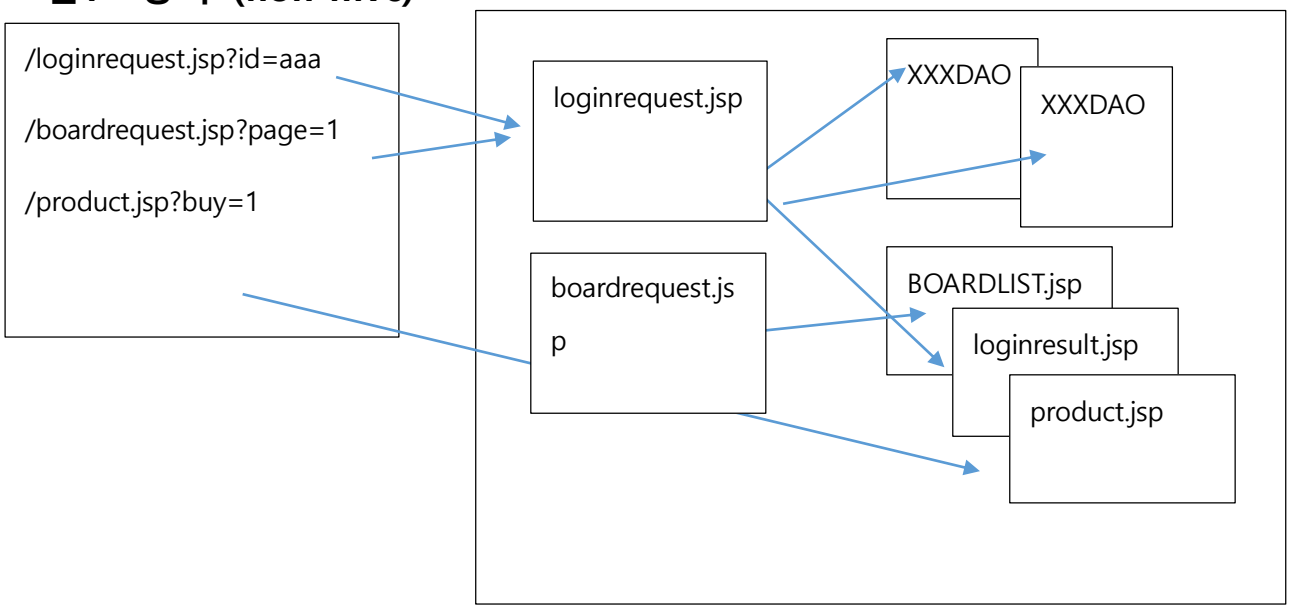
21장 spring mvc

- mvc 란

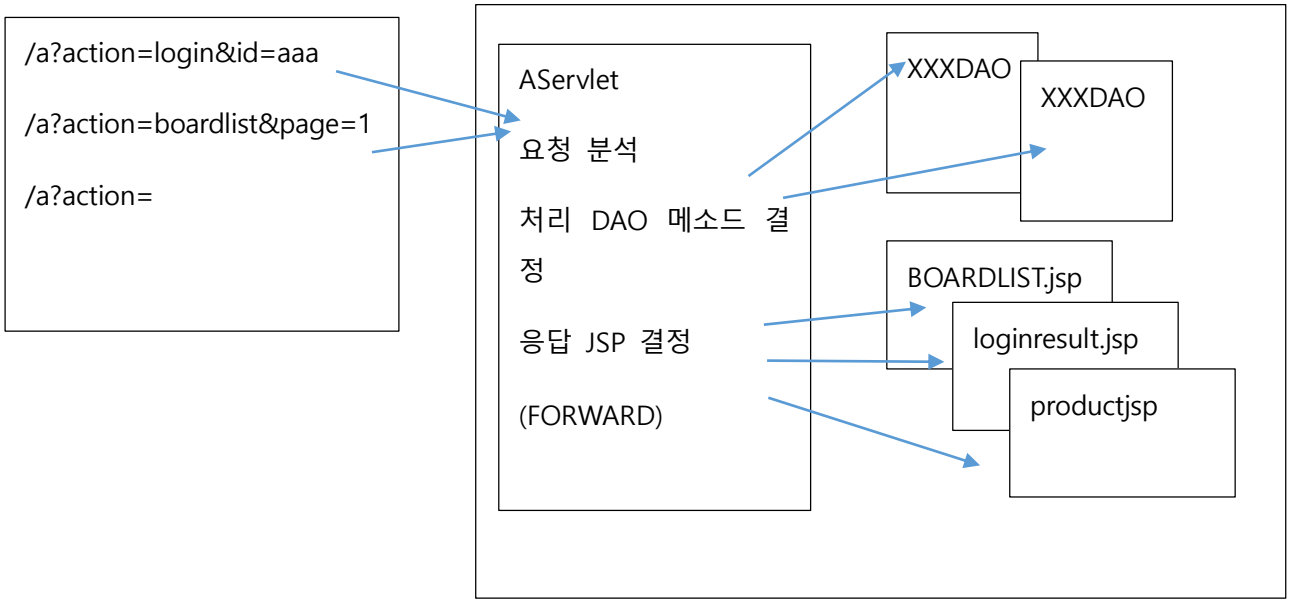
web server – servlet, jsp(html포함), dao, vo

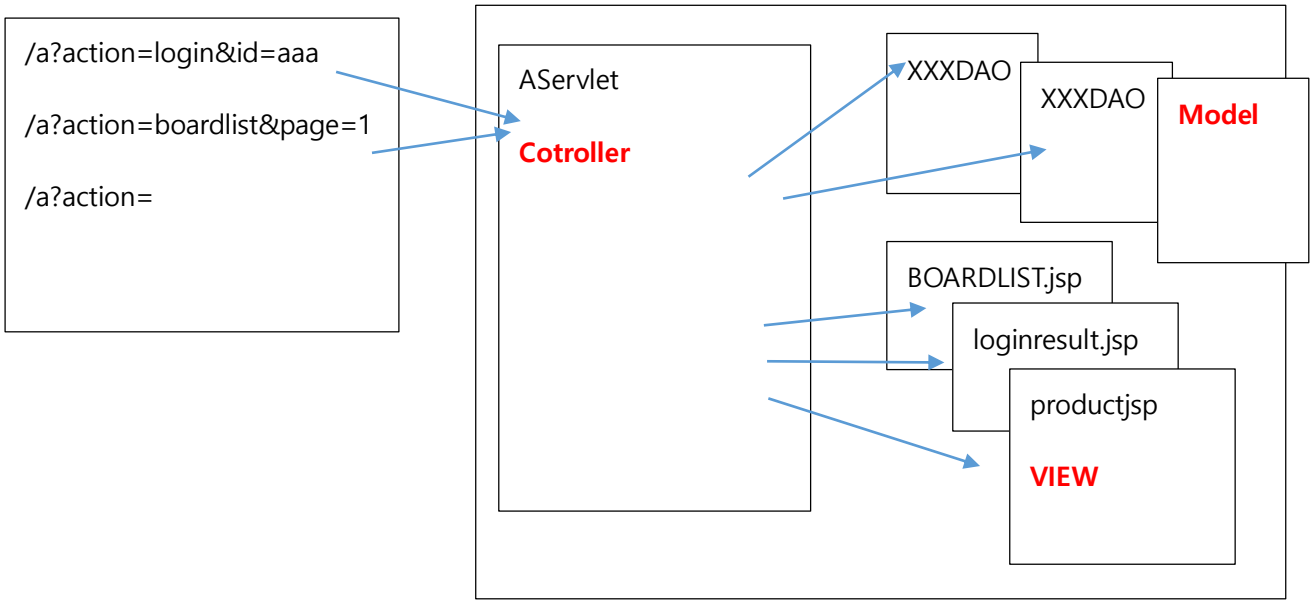
웹서버 개발 형식= 모델1 방식=흐름 복잡해질수록 서버 네부 jsp 피일관계 분석 어렵다

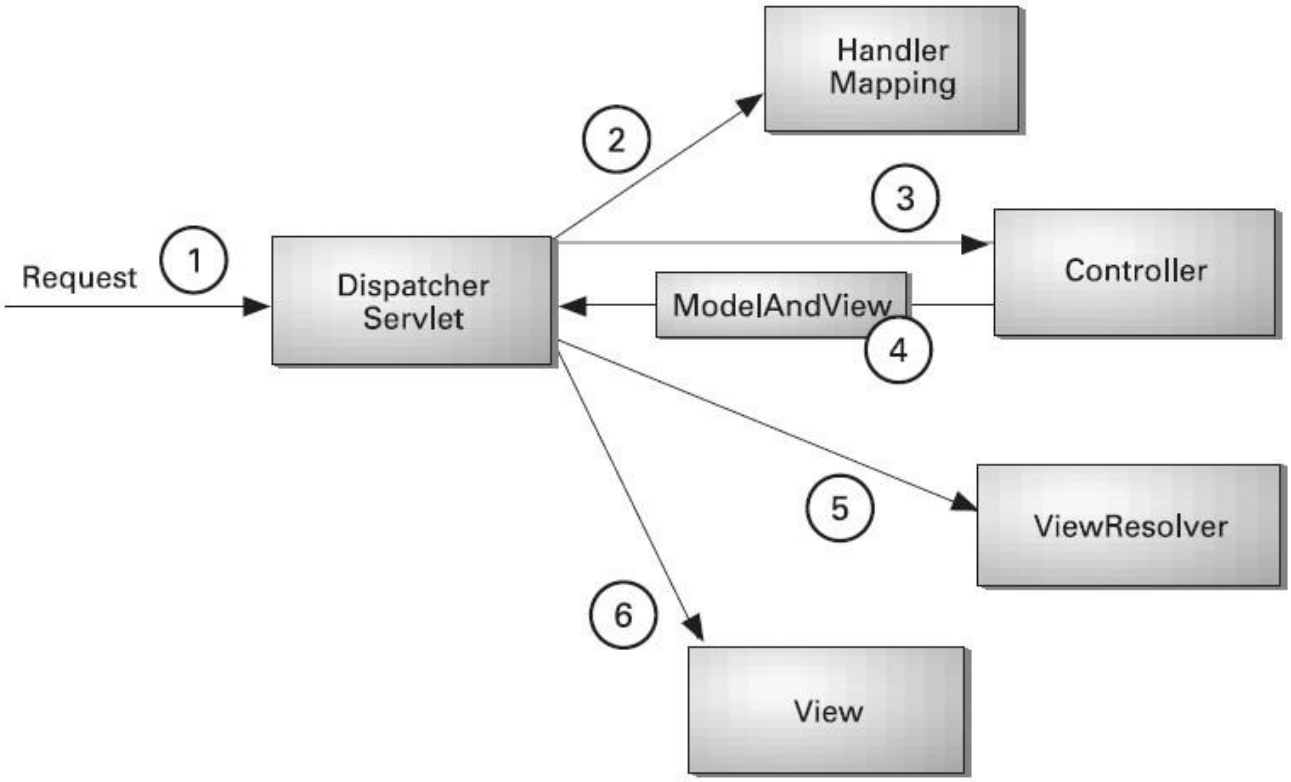
모델1 방식 (non-mvc)



모델2 방식=servlet+jsp 구조




/a?action=login&id=aaa
/a?action=boardlist&page=1
/a?action=
AServlet
Cotroller
XXXDAO
XXXDAO
Model
BOARDLIST.jsp
loginresult.jsp
productjsp
VIEW



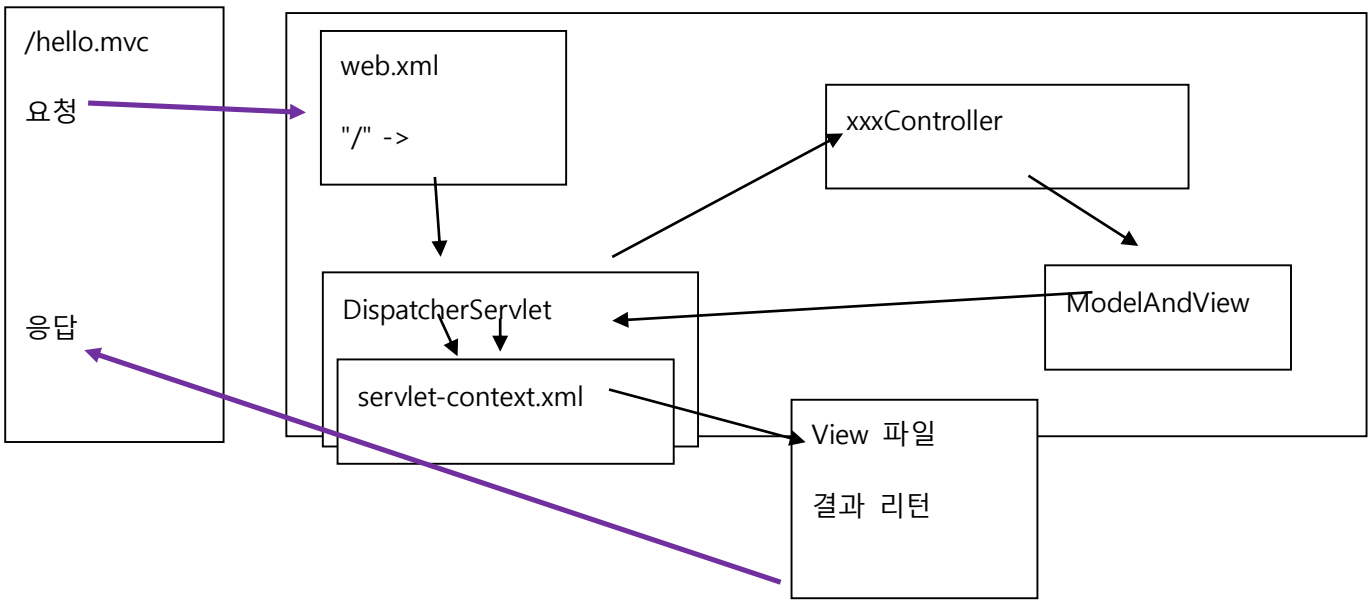
Spring MVC 내에서 처리되는 하나의 요청에 대한 Life Cycle 과정은 위 그림과 같다.

1. 클라이언트의 요청에 대한 최초 진입 지점은 DispatcherServlet 이 담당하게 된다. 대부분의 MVC 를 지원하는 프레임워크가 메인 Servlet 을 가지는 것처럼 Spring MVC 또한 메인 Servlet 이 최초 진입지점으로 다음의 작업을 처리하게 된다.
2. DispatcherServlet 은 Spring Bean Definition 에 설정되어 있는 Handler Mapping 정보를 참조하여 해당 요청을 처리하기 위한 Controller 를 찾는다.
3. DispatcherServlet 은 선택된 Controller 를 호출하여 클라이언트가 요청한 작업을 처리한다.

4. Controller 는 Business Layer 와의 통신을 통하여 원하는 작업을 처리한 다음 요청에 대한 성공유무에 따라 ModelAndView 인스턴스를 반환한다. ModelAndView 클래스에는 UI Layer 에서 사용할 Model 데이터와 UI Layer 로 사용할 View 에 대한 정보가 포함되어 있다.

5. DispatcherServlet 은 ModelAndView 의 View 의 이름이 논리적인 View 정보이면 ViewResolver 를 참조하여 이 논리적인 View 정보를 실질적으로 처리해야할 View 를 생성하게 된다.
6. DispatcherServlet 은 ViewResolver 를 통하여 전달된 View 에게 ModelAndView 를 전달하여 마지막으로 클라이언트에게 원하는 UI 를 제공할 수 있도록 한다. 마지막으로 클라이언트에게 UI 를 제공할 책임은 View 클래스가 담당하게 된다.

"/"===>http:localhost:9090/컨텍스트명/*.*



26장 annotation(xml 태그 + annotation)

ioc di	@Component @Service @Repository @Autowired @Qualifier
mvc	@RequestParam-매개변수 @RequestMapping-메소드 @ModelAttribute-매개변수 @Controller-컨트롤러클래스

- sts에서 컨텍스트명

dynamic web project명	http://ip:port/dynamic web project명
spring mvc project명 package 명 (HomeController.java 제공받을 패키지명) 예: edu.spring.multi	http:ip:port/multi

xml태그	@
A.java class A implements Controller ModelAndView handleRequest(매개변수..) spring mvc 설정 파일 (webapp₩WEB-	A.java @Controller class A @RequestMapping("/xxxx") void m1(){} @RequestMapping("/xxxx") int m2(HttpServletRequest...){}

INF₩spring₩appServlet₩servlet-context.xml) <beans:bean 객체생서... <....	spring mvc 설정 파일 (webapp₩WEB-INF₩spring₩appServlet₩servlet-context.xml) <context:compo..... base-p="패키지명"

-http 오류코드

400	controller @RequestMapping("//) 메소드(int id) 매핑 메소드 매개변수가 form 전달 데이터 타입과 일치하지 않을 때 "111a1"===> int 파일 업로드 – 파일타입 인식 x
404	spring @RequestMapping, @Controller 확인 jsp 존재 확인
405	서블릿 get->doPost 메소드 post->doGet 메소드 jsp get/post 동일 컨트롤러 get/post 나누어 처리
500	자바 예외들
200, 300	정상실행

컨트롤러 메소드의 매개변수 자유롭다

서블릿 api	HttpSession ... 그대로 사용
자바 데이터 타입의 변수	요청 파라미터(input name="xxx") 이름과 동일 이름 변수 자동 저장. 타입 변환 자동. @RequestParam("요청파라미터이름")
자바 객체 form input 모든 요청 파라ㅁ;터값들을 객체 필드변수명 동일 매핑	@ModelAttribute("JSP이름") LoginVO

컨트롤러 메소드의 리턴 타입 자유롭다

ModelAndView	Model, view 이름 설정
void	@RequestMapping("uri") uri 같은 이름의 뷰 설정 model 없다
String	model 없다 뷰이름 설정 return "a";--> WEB-INF/views/a.jsp rerurn "redirect:/a" --> 컨트롤러 @RequestMapping("/a") 메소드 호출 이동

spring mvc 실행 흐름

http://ip:port/context/a.spring (.... view : ab.jsp) --> 404

1. web.xml	"/" –DispatcherServlet 기본 설정 (..servlet-context.xml 설정 참고) 요청값 한글인코딩설정
servlet-context.xml	1> view 저장경로 / 확장자 .jsp 기본 설정 2> @xxx 설정 인식 패키지 설정 <context:component-scan base-pa....="aa.bb.cc"

xxxxx	

-mvc 실습

http://localhost:9090/multi/memberlist	
MemberController ? getMemberList(?) 5개 MemberVO 객체 생성 list 저장 memberlist.jsp 뷰 설정 – list 전송	memberlist.jsp <%@ taglib prefic="c" uri=http://java.sun.com/jsp/jstl/core %> el과 jstl 사용 테이블 태그 5행 4열

```
ArrayList<MemberVO> list = new ArrayList<MemberVO>();
list.add(new MemberVO("member1", 1111, "김회원", "kim@mul.com"));
list.add(new MemberVO("member2", 1111, "박대한", "kim@mul.com"));
list.add(new MemberVO("member3", 1111, "김민국", "kim@mul.com"));
list.add(new MemberVO("member4", 1111, "홍길동", "kim@mul.com"));
list.add(new MemberVO("member5", 1111, "최회원", "kim@mul.com"));
```

21장 mvc
26장 @annotation

- get / post
http 요청 클라이언트 방식

get	post
<form action="/..." method="get" <form action="/..." <a href="/컨트롤러매핑url" 주소입력	html파일 내부 <form action="/..." method="post"

- resources 폴더 사용

src/main/java	컨트롤러 패키지명.xxxx (@ 인식 설정 필요)
webapp/WEB-INF/views	폴더.*.jsp
webapp/resources (mvc 무관 실행 가능)	html *.css *.js jquery-3.2.1.min.js image

- 비지니스로직과 +main/mvc

요청 – 처리(자바 수행 결과) – 응답

1>web.xml

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

classpath:annotation/service/member/member.xml
     </param-value>
</context-param>
```

2> MemberServiceController

@Autowired

MemberService service;

@XXX
메소드(){
?????=service.xxxx()

}

controller – service – dao - vo
- model – view

21장 mvc 기본 기능
26장 annotation
27장 sts4 설치 설정 방법 – sts 3,4 (eclipse+sts....)
28장 mvc 다양한 기능
- file upload 기능

서버 이력서 / 사진 전송 / 상품 이미지 / 게시판 첨부파일 업로드
=== 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.CommonsMultipartResolver" />
```

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

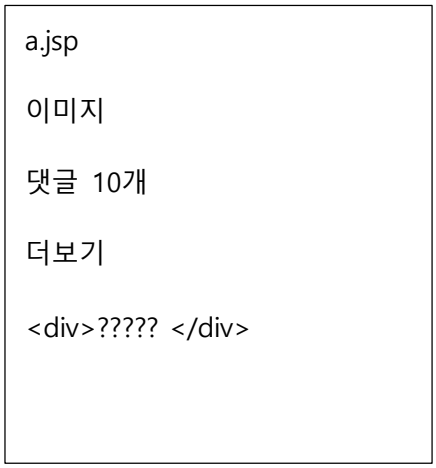
5> MultipartFile 메소드

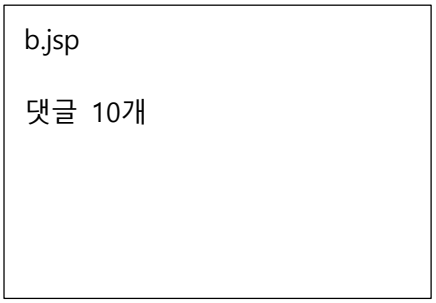
getOriginalFilename()-> 전송파일명

transfer();-> 서버내부 내용 저장

29장 restful 기능 mvc

rest





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
}

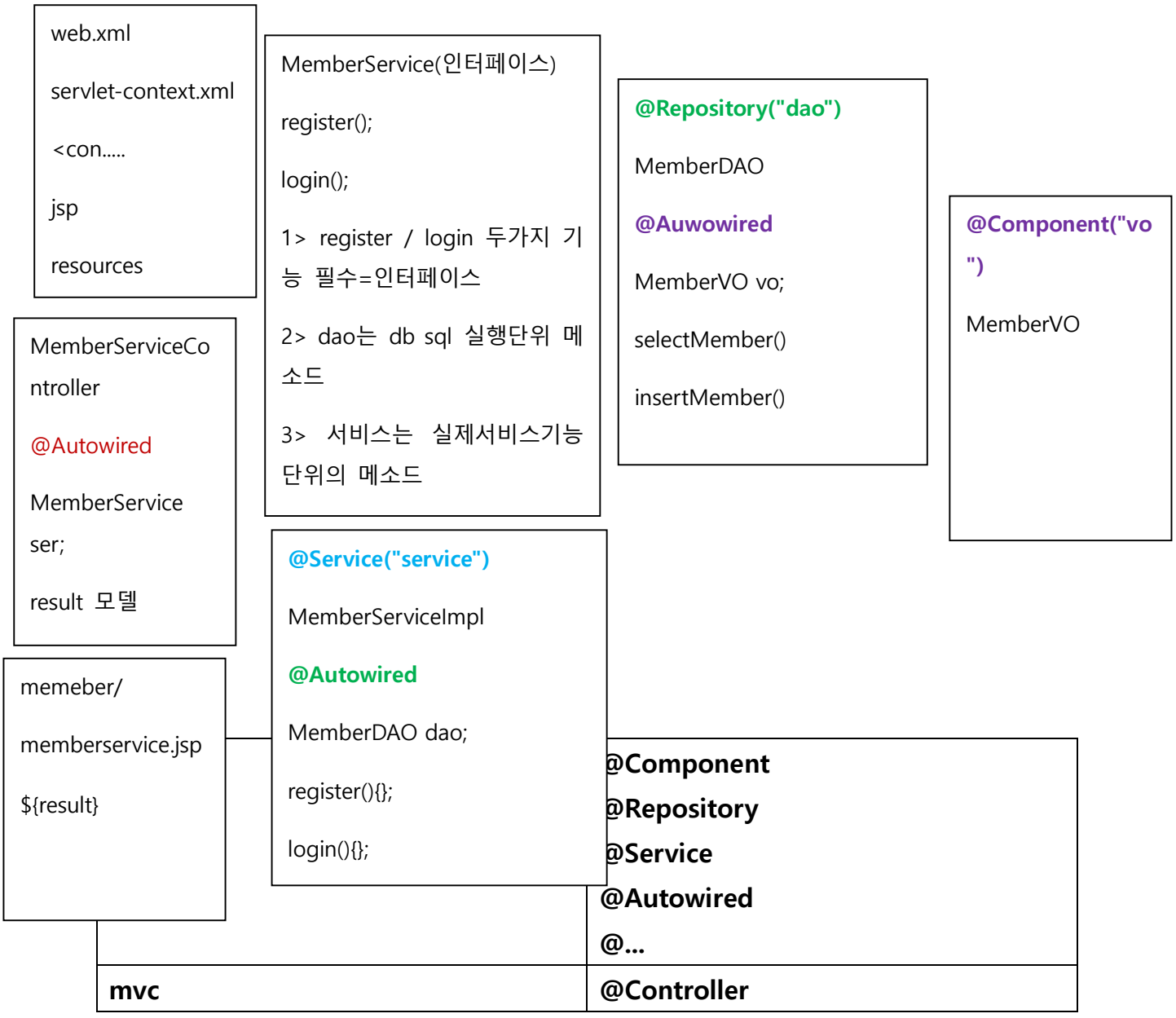
1> pom.xml
ajax하려면 우리가 컨트롤러에서 리턴하는 자바 객체를 자동으로 json 으로 변환해주는 라이브러리 필요

22,23,24장 mybatis 프레임워크 - jdbc 대신 기술

xml 태그방식	annotation 방식

<bean id="a" class="ss.AVO	**@Component("a")**
<bean id="a" class="ss. ServiceImpl"	**@Service("a")**
<bean id="a" class="ss.ADAO"	**@Repository("a")**
<property	**@Autowired**
<property	**@Qualifier**
<constructor-arg	

- xml + annotation



	@Component **@Repository** **@Service** **@Autowired** **@...**
mvc	**@Controller**

	@RequestMapping **@RequestParam** **@ModelAttribute**