

spring framework 특징

- 1> 경량의 컨테이너(여러개 jar 파일 - ioc, di 기능 선택)
- 2> pojo 클래스 사용(dao, vo, service 자바 클래스) 그대로
- 3> framework = 반제품 + 나머지 반(개발내용) = 완제품 . 를 구성
- 4> ioc di 기능으로 스프링은 객체를 생성 - main은 생성된 객체 전달

inversion of control

dependency injection- 객체 외부 전달

4-1. 생성자 통해 객체 주입- 생성자 di

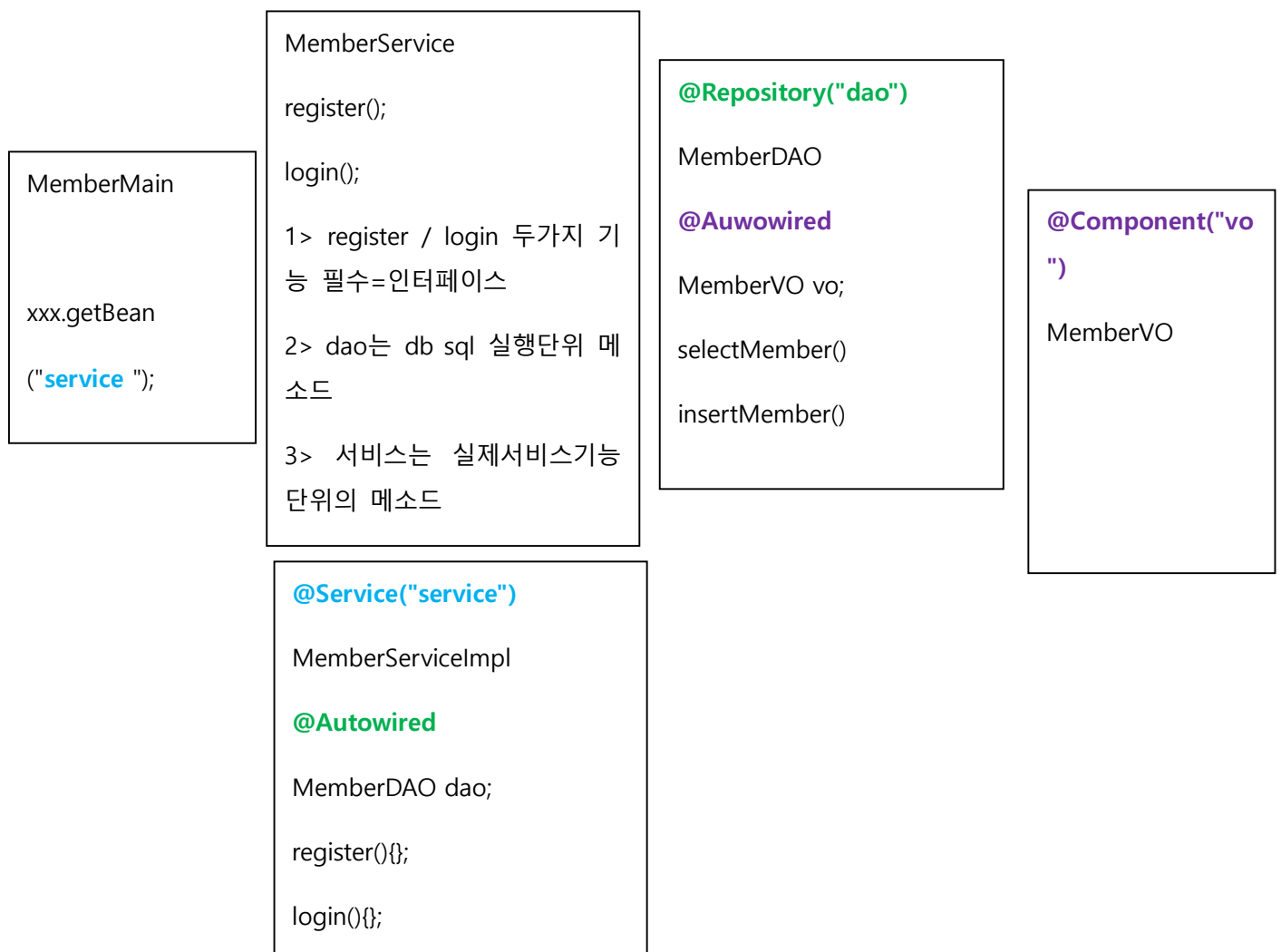
```
class Main{  
    B b1 = new B();  
    B b2 = new B();  
    A a1 = new A(b2);  
}  
class B{}  
class A{  
    B b1 ;  
    A(B b1){  
        this.b1 = b1;  
    }  
}
```

4-1. setter 메소드 주입- setter di

```
class Main{  
    B b1 = new B();  
    B b2 = new B();  
    A a1 = new A();  
    a1.setB(b2);  
}  
class B{}  
class A{  
    B b1;  
    A(){  
        setB(B b1){  
            this.b1 = b1;  
        }  
    }  
}
```

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



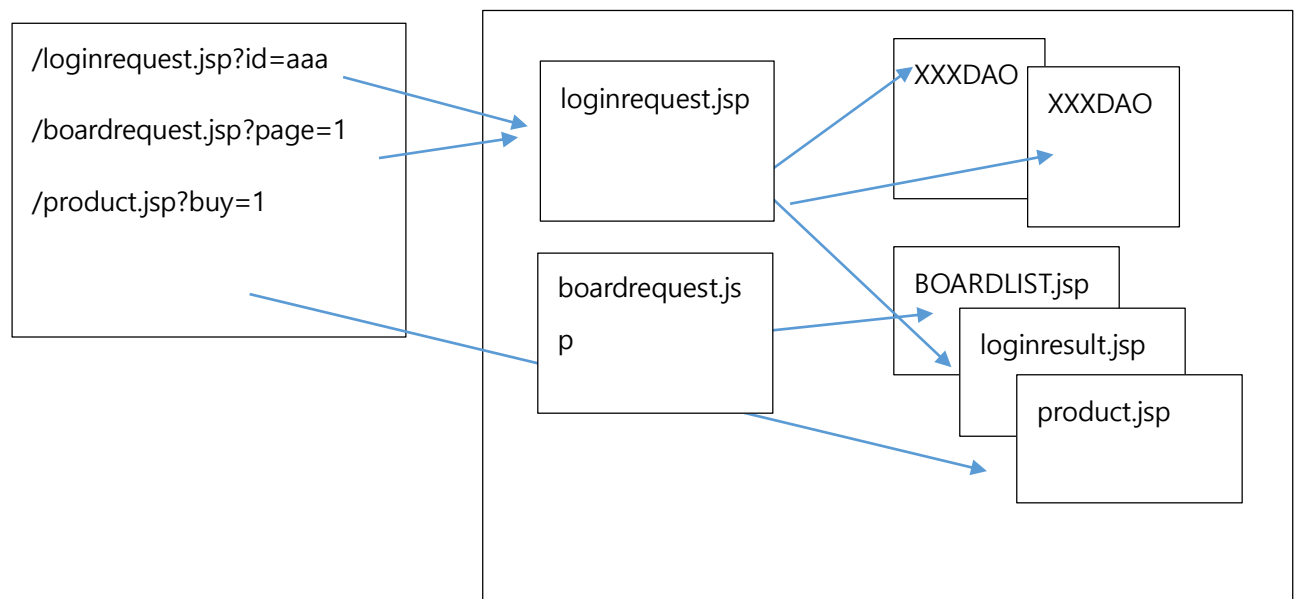
aop	mvc	spring jdbc	프레임워크연동
기본 필수 ioc di			

21장 spring mvc

- mvc 란

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

웹서버 개발 형식= 모델1 방식=흐름 복잡해질수록 서버 내부 jsp
파일관계 분석 어렵다

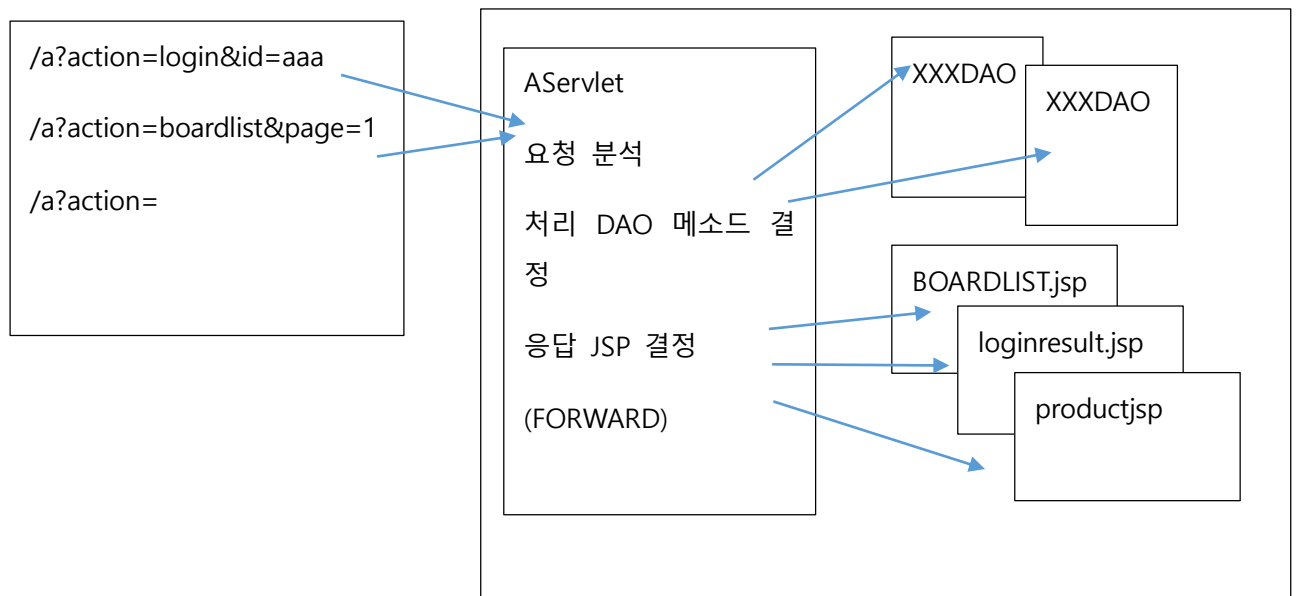


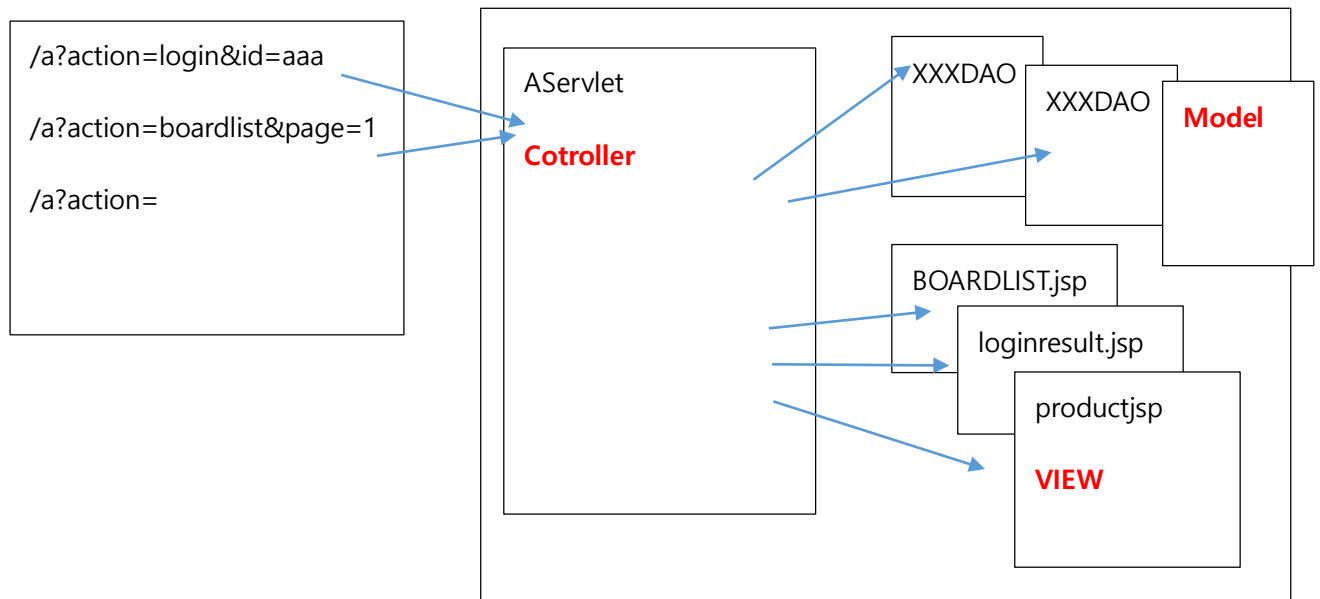
servlet – 클라이언트 요청 – 분석 - 자바 로직 처리 (dao호출)

dao ,vo - 실제 처리 결과물 생성-저장

jsp – 응답

모델2 방식=servlet+jsp 구조





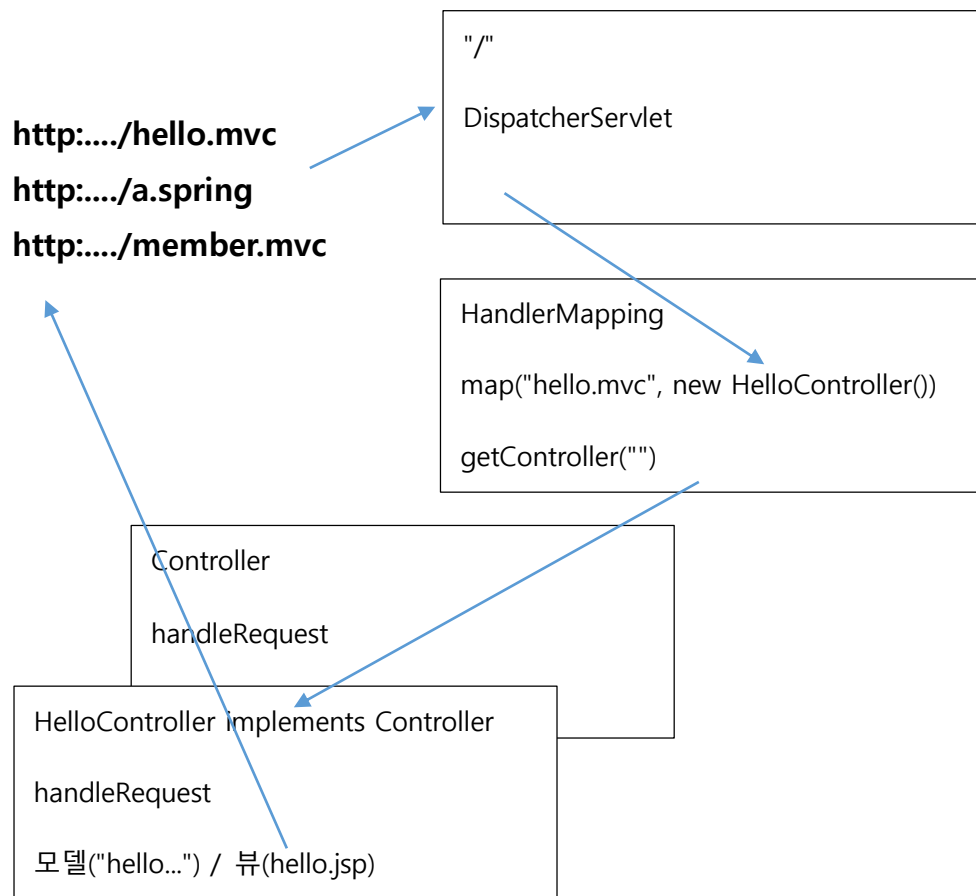
MVC 우리 구현

===> **spring mvc**

강제화

FRONT CONTROLLER + MVC

- MVC – DYNAMIC WEB PROJECT



DispatcherServlet	클라이언트 모든 요청 집중 ("/") frontcontroller역할 요청 가장 먼저 받아서 view 이동
HandlerMapping	특정 url – 컨트롤러 매핑 정보 hello.spring-new HelloController() a.spring – new AController()
Controller	메소드
HelloController AController	특정 url – model, view 설정 model 설정
hello.jsp a.jsp	응답

spring mvc (스프링 프로젝트 폴더 구조 다르다)

DispatcherServlet	x-spring api
HandlerMapping	x-spring api
Controller	x-spring api
HelloController AController	src/main/java
hello.jsp a.jsp	src/main/webapp/WEB-INF/views
dao, vo	src/main/java
spring mvc 설정 xml	src/main/webapp/WEB-INF/app-Servlet

26장 annotation(xml 태그 + annotation)

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

dynamic web project이름 = http://ip:port/ dynamic web project이름

sts4 툴

spring project 이름(spring_first) ==>

package 이름 (HomeController.java 제공받을 패키지)

==> edu.spring.**multi**

http:ip:port/**multi**

xml태그	@
A.java class A implements Controller ModelAndView handleRequest(매개변수..) spring mvc 설정 파일 (webapp\WEB-INF\springWappServlet\servlet- context.xml) <beans:bean 객체생성... <....	A.java @Controller class A @RequestMapping("/xxxx") void m1(){} @RequestMapping("/xxxx") int m2(HttpServletRequest...){} spring mvc 설정 파일 (webapp\WEB-INF\springWappServlet\servlet- context.xml) <context:compo..... base-p="패키지명"

CREATE / READ / UPDATE / DELETE

==> CRUD 기능 수행 메소드들 --> 1개 컨트롤러

@Controller

```

class BoardController{
    @RequestMapping("/write.board")
    글저장메소드(){}
    @RequestMapping("/detail.board")
    글상세조회메소드(){}
    글리스트메소드(){}
    글수정(){}
    글삭제(){}
}

```

LoginController

login.html ---> LoginServlet --> 이동 ---> loginresult.jsp
form 입력-처리

<http://localhost:9090/multi/loginform> -->

loginform.jsp ---> /loginresult 전송

--> DispatcherServlet

--> servlet-context.xml (매핑 , @.....)

@Controller

class A

@RequestMapping("/loginform")

login 메소드 실행

return "loginform";

--> /web-inf/views/loginform.jsp

@RequestMapping("/loginresult")

loginresult 메소드 실행

.....처리 결과 "loginresult"

400	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	
500
200, 300	정상실행

컨트롤러 메소드의 매개변수 자유롭다

서블릿 api	HttpSession ... 그대로 사용
자바 데이터 타입의 변수	요청 파라미터 이름과 동일 이름 변수 자동 저장. 타입 변환 자동. @RequestParam("요청파라미터이름")
자바 객체	@ModelAttribute("JSP이름") LoginVO

컨트롤러 메소드의 리턴 타입 자유롭다

서블릿 api	HttpSession ... 그대로 사용
자바 데이터 타입의 변수	요청 파라미터 이름과 동일 이름 변수 자동 저장. 타입 변환 자동. @RequestParam("요청파라미터이름")
자바 객체	@ModelAttribute("JSP이름") LoginVO

spring mvc

web.xml	"/" -DispatcherServlet (..servlet-context.xml 설정 참고) 요청값 한글인코딩설정
servlet-context.xml	view 저장경로 / 확장자 @xxx 설정 인식 패키지 설정
XXXXXX	

http://localhost:9090/multi/memberlist	
MemberController ? getMemberList(?) 5개 MemberVO 객체 생성 list 저장 memberlist.jsp 뷰 설정 – list 전송	memberlist.jsp <%@ taglib prefix="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"));

```