

- spring ioc와 di 설정

외부로부터 객체 전달 개념

ioc 표현 스프링

setter injection

constructor injection

스프링에서 생성 자바객체=pojo=bean-생성-공유-소멸 관리

spring 설정 xml과 annotation

| 19장 ioc di xml방식  | 26장 annotation 방식   |
|---|---|
| <bean id="a" class="ss.A"   | <b>@Component("a")</b><br>class A {}  |
| <bean id="a" class="ss.xxxServiceImpl"  | <b>@Service("a")</b><br>class xxxServiceImpl implements xxxService {}   |
| <bean id="a" class="ss.ADAO"><br><property name="vo" ref="v" /><br></bean><br><bean id="v" class="ss.AVO"   | <b>@Repository("a")</b><br>class ADAO {<br><b>@Autowired</b><br>AVO vo;<br>}<br><b>@Component("vo")</b><br>class AVO{}<br><br><b>@Repository</b><br>class BDAO {} |
|   | <b>@Autowired</b>   |
| <b>@Service</b><br>class ServiImpl{<br><b>@Autowired</b><br><b>@Qualifier("aDAO")</b><br>ADAO dao;<br>}<br><br><b>@Repository("aDAO")</b><br>class ADAO{<br>} | <b>@Qualifier</b>   |

```
xml <bean id="dao2" class="ADAO"
```

## - 21장 spring mvc

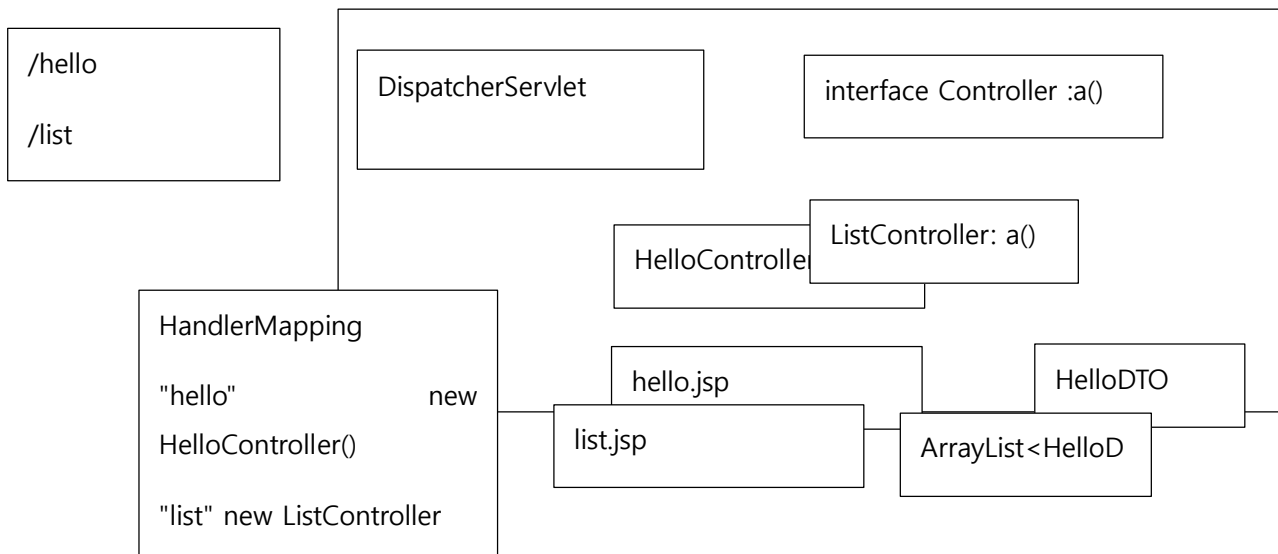
### 웹페이지개발 – 서버 자바

### SERVLET JSP XXXDAO, XXXDTO

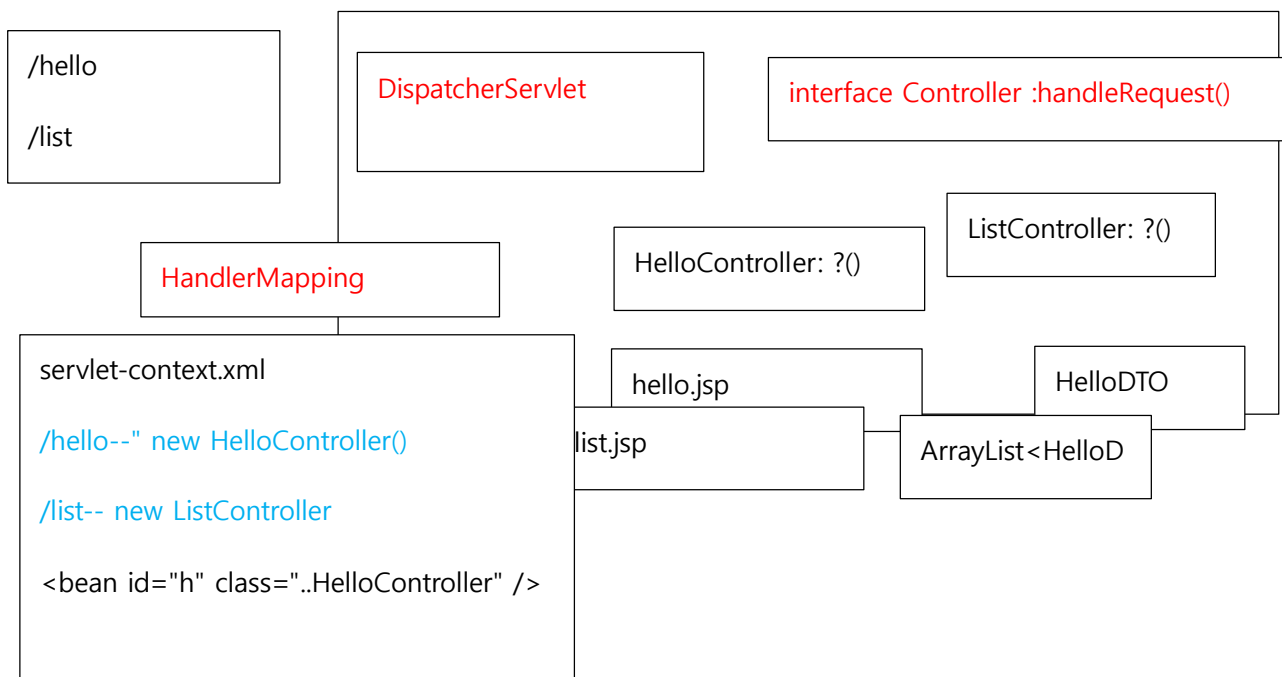
### 요청- FRONTCONTROLLER 서블릿 - C – M – V -응답

/list

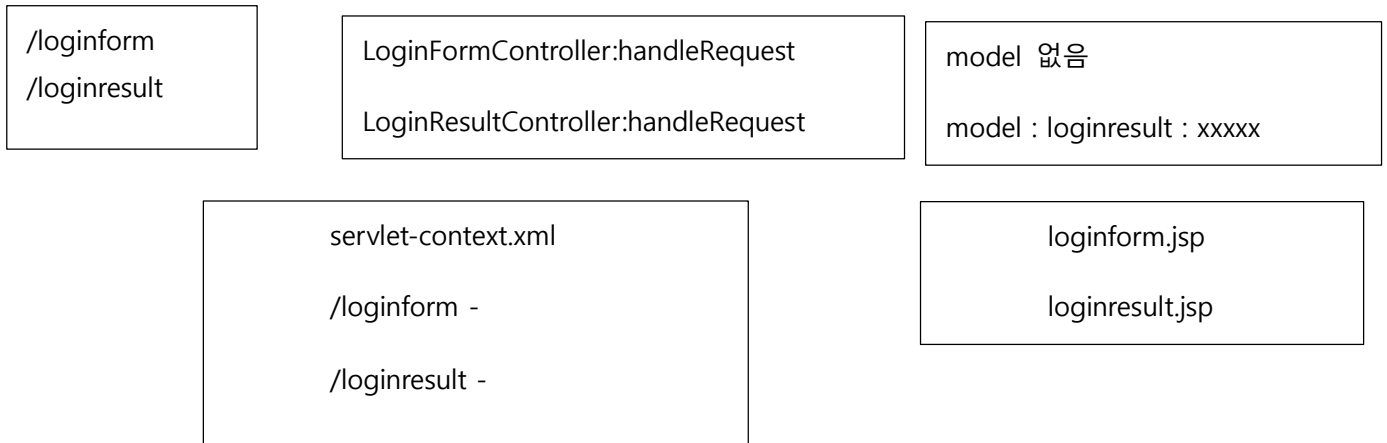
#### 1. non-spring mvc- dynamic web project



#### 2. spring mvc 변경- spring mvc project



- spring mvc 로그인처리

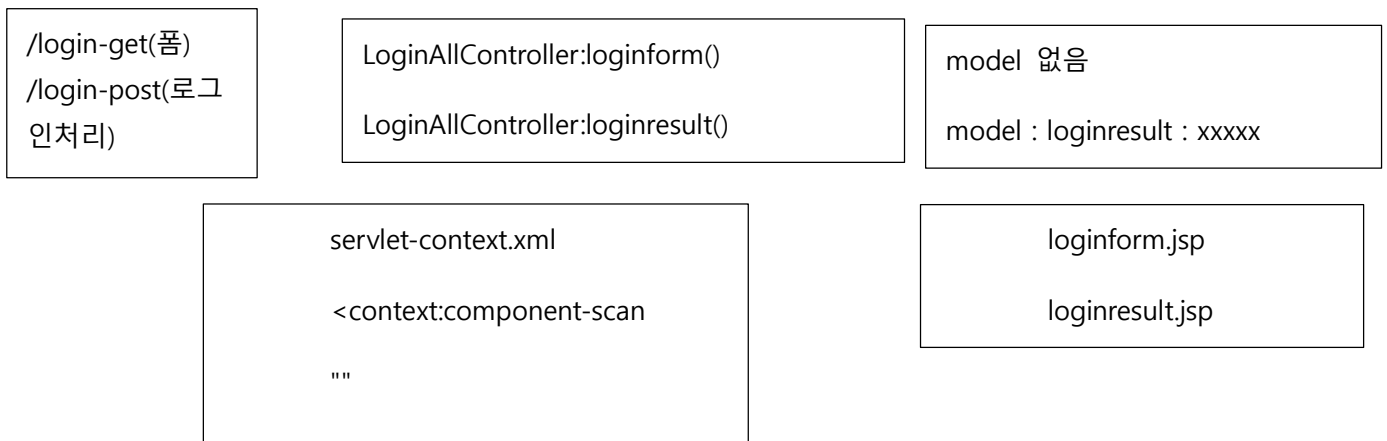


- spring mvc 의 컨트롤러

- 1> mvc 형태
- 2> /WEB-INF/views/\*.jsp 저장 설정
- 3> Controller 상속 handleRequest메소드 구현
- 4> 리턴타입 ModelAndView
- 5> 매개변수 HttpServletRequest.....
- 6> 1개 url 매핑 1개 메소드만 실행

|   |   |
|---|---|
| spring mvc<br>xml 선언<br>class A implements Controller<br>리턴타입 handleRequest(매개변수)..<br><br><bean ..<br>id="hello" class="HelloCONTROLLER" /><br><br><....key="/hello"> hello /> | SPRING MVC ANNOTATION<br>@Controller<br>class A<br>@RequestMapping("/a",method=get/post)<br>@GetMapping("/a") -> 4.3 이후 추가<br>public ? m(){<br>@PostMapping |
|---|---|

- spring mvc 로그인처리



로그인

회원가입

(2개- 폼 출력 화면 + 사용자입력내용처리결과 출력 화면)

|                                      |  |
|--------------------------------------|--|
| @Controller                          | 컨트롤러 클래스 위 선언<br>@Component @Repository @Service @Controller |
| @RequestMapping("/a") - get 그리고 post | 메소드 위 선언   |
| @GetMapping("/a") - get              | 메소드 위 선언   |
| @PostMapping("/a") - post            | 메소드 위 선언   |
| @RequestParam("요청파라미터명") String 변수   | 매핑 메소드 매개변수앞 선언  |
| @ModelAttribute("모델명") xxxDTO dto    | 매핑 메소드 매개변수앞 선언  |

작성하실 때 getmapping은 "login"이고 postmapping은 "/login"으로 차이가 나는 이유가 따로 있을까요?

1>html 태그 "포트번호/..."

2>서블릿, jsp, spring mvc "컨텍스트명/..."

3> 서블릿, jsp, spring mvc "현재파일과 같은 url기준."

- 컨트롤러 메소드 매개변수

1> 없어도 된다

2> 요청 처리 데이터만 전달

3> HttpServletRequest 서블릿 api

4> 자바 기본형변수나 문자열, 배열 변수 선언

변수이름= 요청파라미터이름 저장

5> 자바 객체 변수 선언

객체 내부 멤버변수이름=요청파라미터이름 저장

- 컨트롤러 메소드 리턴타입

|              |  |
|--------------|--|
| ModelAndView | 모델과 뷰 설정                                 |
| String       | 뷰이름 설정(모델 없거나 @ModelAttribute("dto") 대신) |
|              |  |
| void         | 뷰이름은 규칙대로.                               |

|     |  |
|-----|--|
| 404 | 요청url 파일 없다<br><a href="http://ip:port/mvc/a">http://ip:port/mvc/a</a><br><br>/a 매핑 컨트롤러 메소드 없다-@RequestMapping<br>/a 매핑 컨트롤러 메소드는 있으나 view 없다 |
|-----|--|

|     |   |
|-----|---|
| 405 | http 요청방식 안맞는다<br>@PostMapping("/a")....<br><form action="a" method=get >   |
| 500 | 서버 예외   |
| 400 | 요청파라미터 타입 안맞는다<br>/a?age=이십세<br><br>@RequestMapping("/a")<br>..메소드(int age) |

/memberinsert-get

/memberinsert-post

annotation.test.MemberController

/memberinsert-get 메소드- memberinsert.jsp 뷰

/memberinsert-post 메소드 – MemberDTO 타입 저장. member.jsp  
출력 뷰

memberinsert.jsp

id password name phone email address 입력폼

회원가입 클릭하면 memberinsert-post 전달 정의

member.jsp

id password name phone email address 출력