

스프링 기술

aop	mvc	spring jdbc	프레임워크연동
기본 필수 ioc di			

26장 annotation(xml 태그 + annotation)

ioc di	@Component @Service @Repository @Autowired @Qualifier
mvc	@RequestParam -매개변수 @RequestMapping -메소드 @ModelAttribute -매개변수 @Controller -컨트롤러클래스 @ResponseBody -ajax 리턴 json 객체

- get / post

http 요청 클라이언트 방식

get	post
<pre><form action="/..." method="get" <form action="/..." <a href="/컨트롤러매핑url" 주소입력 @Controller class A @RequestMapping(value="/xxx", method=RequestMethod.GET) ? m(???){} ? m(???){} </pre>	<pre>html파일 내부 <form action="/..." method="post" @Controller class A @RequestMapping(value="/xxx", method=RequestMethod.POST) ? m(???){} </pre>

- resources 폴더 사용

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

- 비즈니스로직과 + mvc

web.xml	<pre> <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> </pre>
xxxxController	<pre> @Autowired xxxxService service; </pre>

- file upload 기능

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

=== file upload

1>클라이언트 html 태그 업로드 기능

<form action="..."

method="post"

enctype="multipart/form-data"

<input type="file" --> 파일선택열기창

=====

서버 파일 전송받는 기능-spring api 추가 (파일명+내용+....)

2> pom.xml – 27 장

mvnrepository.com

spring 라이브러리 다운로드 사이트 여러 군데 나누어졌다

통합 버전 관리 사이트

라이브러리이름, 버전 선택 – tag 복사

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 ajax와 json

1> 기기 다양- 폰, 패드, 노트북, pc 다양한 화면 크기

화면 너무 많은 내용 들어가면

2> map -

3> 기본 화면 구성 + 추가 내용 변경 동적

```
a.jsp  
  
이미지  
  
댓글 10개  
  
더보기  
  
<div>????? </div>
```

```
b.jsp  
  
댓글 10개
```

http 특징

1> 요청 - 응답

2> 기본적 요청 처리 시간 동안 클라이언트 대기한다. 화면 아무 변화없다.
응답 후에 전체 화면은 변경된다.

---> 동기화 통신 방식(synchronous)

ajax 특징

3> 화면 전체 구성하고 요청 - 응답 - 화면 일부 변경

---> 비동기화 통신 방식(asynchronous)

--> java script and

--> xml (비동기통신시 전달 데이터형태 초기)

--> 자바 스크립트 객체(비동기통신시 전달 데이터형태 현재)

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

--> json --> java script object notation

--> <div>,

@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

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

spring xml 파일들

pom.xml	spring mvc project root 스프링 라이브러리 다운로드, 버전 설정
web.xml	src/main/webapp/WEB-INF/ 웹 관련 설정 (url – tomcat – 스프링 이전 전달 정보들) 1>url -->스프링 DispatcherServlet 2>encoding utf-8 설정 3> 스프링 전달 추가 설정xml 정보
servlet-context.xml	src/main/webapp/WEB-INF/spring/appServlet 스프링 설정 1>view 경로와 확장자 2>정적 요소들 - html css js image 경로 3><context:component-scan
	src/main/java-->*.java
추가 설정xml	member.xml +

28장 file upload

29장 ajax

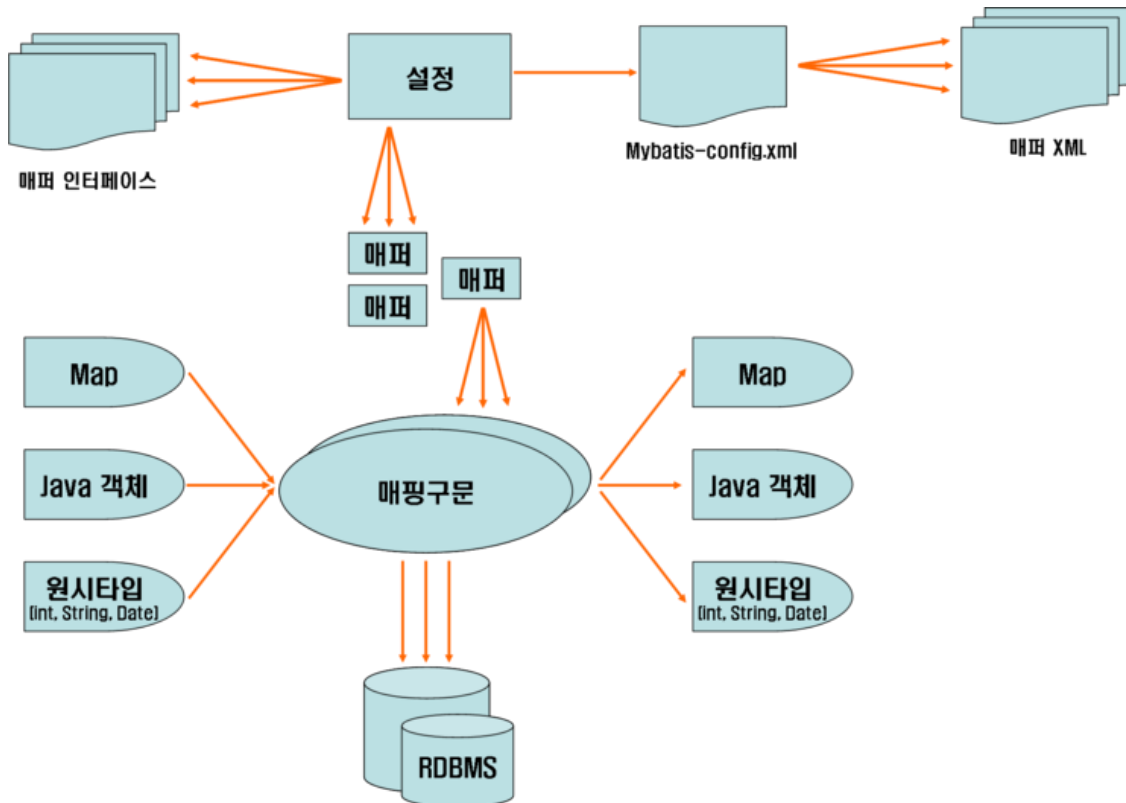
스프링	mybatis
자바	자바
통합 기능 ioc di + mvc + aop + db 연동 + 다른 프레임워크 연동 dao, vo 코드 그대로 =비지니스로직	db 연동 기능

23장 mybatis

jdk jdbc	spring jdbc	mybatis
java.sql.*	스프링제공jdbc	jdbc 프레임워크
1>코드 반복 심하다 try{ Class.forName("") DriverManager.... ("", "", "") Result<=- pt.executeQuery("select") vo.setPassword (rs.getInt("컬럼")) int <_pt.executeUpdate() con.close() }catch }finally 2>connectionpool 수동 선택 연결 유지 tomcat		1> xml 설정 sql 문장 자바 코드 제거 ==> xml 별도 설정 2> connectionpool, sql실행 객체 자동 생성 - 관리 3> sql 조회 결과를 String/MemnerVO/ArrayList 정한 타입 설정

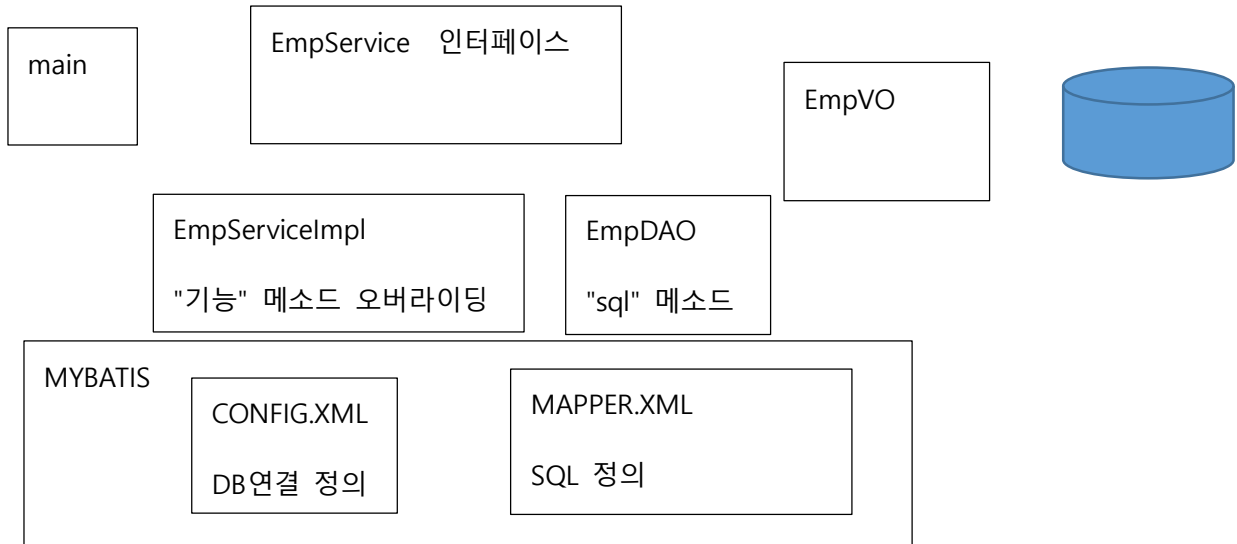
server.xml		
------------	--	--

- 객체지향 언어인 자바의 관계형 데이터 베이스 프로그래밍을 좀더 쉽게 할수 있게 도와주는 개발 프레임워크.
- 자바에선 데이터베이스 프로그래밍을 하기 위해 JDBC(자바에서 제공하는 데이터베이스 프로그래밍 API)를 제공
- JDBC는 관계형 데이터 베이스를 사용하기 위해 다양한 API를 제공
- JDBC는 코드 반복이 너무 많다는 단점이 있음
- 처음에는 ibatis 라는 이름으로 시작 ->MYBATIS로 변경



- pom.xml

mybatis+oracle	mybatis.jar ojdbc8.jar
spring+mybatis+oracle	mybatis-spring.jar spring-jdbc.jar



mybatis-config.xml(시작 설정정보)

객체 type 설정	<typeAlias
연결할 DataSource 설정	<environment 다중 환경을 설정할 수는 있지만 SqlSessionFactory 인스턴스마다 한 개만 사용할 수 있다는 것이다.
mapper 파일 설정	<mapper

sql-mapping.xml

select	id="자바 sql 실행 식별자" resultType="1개 레코드 타입" parameterType="자바에서 sql 로 전달 데이터"
insert	id=필수 resultType 없다
update	id=필수

	resultType 없다
delete	id=필수 resultType 없다

resultType	XXXVO int-그룹함수 결과 1개 select String
parameterType	xxxxVO int String int[]

```
select first_name, department_id
from employees
where department_id in (10, 50, 80);
```

```
<select ... parameterType="int, int, int"
select first_name, department_id
from employees
where department_id in ({a}, {b},{c});
==> error
```

mybatis 패키지

MemberMain	MemberService	MemberDAO	MemberVO
member테이블의 email에 'mul' 포함 회원 의 정보 출력 아이디-이름-이메일 형식	MemberServiceImpl		
member- config.xml (mybatis- config.xml 복사 붙여넣기) 1.typeAlias 변경 2.environment 불 변 3. mapper 변경	member- mapping.xml		

24장 spring mybatis 연동

mybatis-spring연동설정xml	
mybatis-config.xml	
mapper xml	
servlet-context.xml	
web.xml	