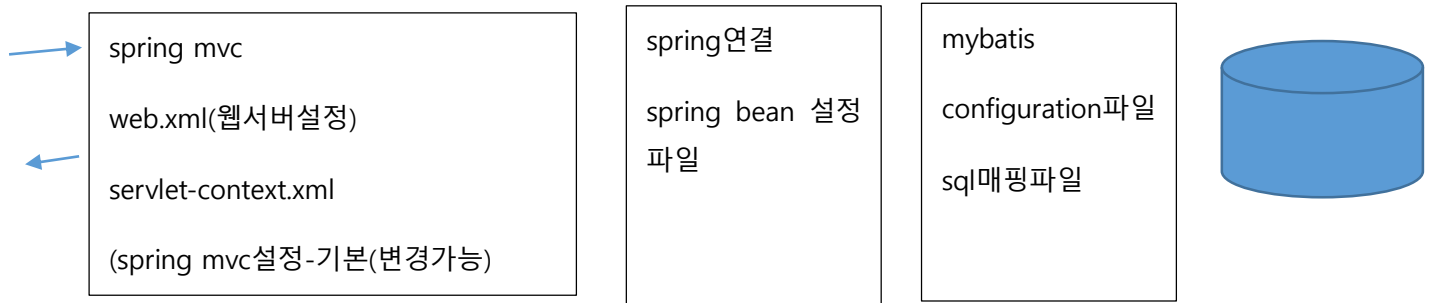


24장

mybatis 프레임워크 - db 연동 프레임워크

spring mvc + mybatis -xml



member_table crud - spring mvc + mybatis

28장

http 요청 파일 서버 전송 기능 없다

HttpServletRequest

getStringParameter("")

getStringValues("")

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

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.

1> pom.xml = maven 다운로드 요청

commons-io.jar

commons-fileupload.jar

<!-- 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>

commons-fileupload-1.3.1.jar 파일 실제다운로드

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

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.6</version>

</dependency>

2> servlet-context.xml (서버로 name=xxxx,..... 설정)

<beans:bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver" />

==> Multipart 객체 생성

3> 코드

client	server
<pre><form action="a" method="post" enctype="multipart/form-data" > <input type="file" name="f"> <INPUT TYPE=TEXT name="id"</pre>	<pre>@Controller class A @RequestMapping("/a") public void m(Multipart f, String id,){</pre>

<input type="submit" value="파일전송"/> </form>	f.getOriginalFileName() f.getLength() f.transfer(다른파일명 서버 저장) }
--	--

-실습

UploadController	@RequesMapping(value="/fileupload", method=....GET) 메소드() 모델 없다 uploadform.jsp 뷰(파일2개 업로드, name, description 입력-전송) } @RequesMapping(value="/fileupload", method=....POST) 메소드(UploadDTO dto){ 파일을 서버 c:\upload 폴더 파일 저장 처리결과 모델 정의 uploadresult.jsp 뷰 }
UploadDTO	폼 입력 모든 내용 저장 변수들 객체 name description file1 file2
uploadform.jsp	파일업로드 폼 생성
uploadresult.jsp	

- 다운로드(라이브러리추가x)

- 서버의 특정 디렉토리는 웹브라우저에서 접근할 수 없다.
- 웹브라우저에서 사용하도록 하려면 다운로드 기능을 제공한다.
- 파일을 읽어 HttpServletResponse의 OutputStream으로 출력하도록 구현한다.
- 파일 다운로드와 관련된 헤더 정보를 설정해준다.

response.setContentType("text/html;charset=utf-8"); 파일다운로드 없을때
out = resonse.getWriter();
out.println("<html"

response.setContentType("application/download");
response.setContentLength((int)파일객체.length());
response.setHeader("Content-Disposition",

```
"attachment;filename=\""+filename+"\"");
```

```
response.getOutputStream();
```

```
pom.xml<dependency >
```

```
servlet-context.xml <context:compo...
```

```
<bean CommonsMultipartResolver
```

업로드파일목록 영구적 저장 mybatis

=====

29장 restful 기능 mvc

rest

ajax 스프링

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)

매번 요청 - 응답(동일-일부 변경) 전체 화면 렌더링한다

spa-single page application

map

asynchronous java script and xml(json) ==> ajax

<bean id="" class="" />

@Component

html태그 불가능하다

jquery 구현

<form action="" method=get|post

클라이언트	spring mvc 서버(ajax처리)
<pre>\$.ajax({ url:서버호출uri(컨트롤러 매핑url) , data: { id:"xxx",pw:1111,... }, type:get post, dataType: 'json' success:function(a){ 서버로부터 비동기적 응답 a.id a.loginresult });</pre>	<pre>@Controller class A { @ResponseBody @RequestMapping("/uri") public String m(String id, String pw){ //처리-service-dao... return "{"id":xxxx, "loginresult":true}"; } json [{ "변수명": "값" , 변수명:true/false, } , { "변수명": "값" , 변수명:true/false, }]</pre>

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

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> src/main/webapp/resources폴더에 jquery 라이브러리 파일 위치

3> \$.ajax / @ResponseBody

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

@ResponseBody	ajax처리결과 메소드 위 선언 리턴타입 자바 객체 - json형태변경
@RestController	컨트롤러클래스 위 선언 모든 메소드 요청 ajax 처리
@PathVariable http://localhost:8080/컨텍스트명 /m/abc/1234 http://localhost:8080/컨텍스트명/m/spring http://localhost:8080/컨텍스트명/m/ajax <form action=" http://localhost:8080/컨텍스트명/m/\${???"	@Controller class A{ @RequestMapping("/m/{id}/{pw}") public void m(@PathVariable("id") String id, @PathVariable("pw") String password){ } }

"	
@RequestParam http://localhost:8080/컨텍스트명 /m?memberid=abc	@Controller class A{ @RequestMapping("/m") public void m(@RequestParam("memberid") String id){ } }

====SPRING LEGACY PROJECT(MVC)====

19 - 29

1>spring ioc, di

2> spring mvc

3> spring 다른 프레임워크 연동(mybatis)

4> mybatis

5> 파일 업로드 / 파일 다운로드

6> SPA-

AJAX+SERVLET+JSP

AJAX=SPRING MVC

STS4	SPRING BOOT 프로젝트만 생성 ("자바" 개발자+xml설정 줄여보자) MVC IOC DI 업로드 AJAX
STS3 ADD ON 추가 설치	SPRING MVC 프로젝트 생성 3.1.1. --> 4.3.18

====SPRING STARTER PROJECT====

=====

32장 스프링 부트

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

2> 스프링 부트 --> 설정 간소

=====