

세미프로젝트

화 - 1시간 인사

수 - 2시간 주제/팀장/메뉴(기능, 서비스), 화면 흐름도, db 구성. 테이블 sql, 초기 테이블 insert sql

목 - 2시간 패키지명.클래스 규칙. 온라인 각자 역할분담 (게시판 / 상품 / 장바구니 - 기능 담당)

금 - 3,4단원 과제 2개, 시험

다음주 월화수 3일 - 세미프로젝트- 주제/설계/각자구현/통합/시연/발표

조편성 - 5/5/6/6

- 28장 다중파일 업로드 / 다운로드

MultipartFile --> 파일 내용, 이름, 길이,,,,여러가지 정보 객체

1> 게시판 (번호,제목,,,,첨부파일)

2> 온라인 이력서 파일 제출

3> 상품이미지사진 관리자 파일

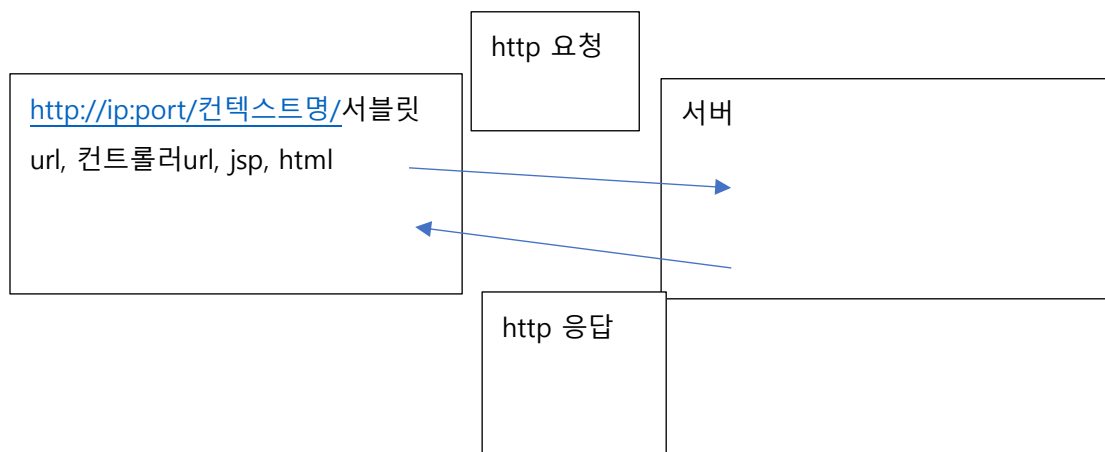
==> 클라이언트에서 서버로 파일 업로드

4> 다운로드

서버에서 클라이언트로 파일 다운로드

파일업로드 - multipart

web http 프로토콜 "요청" 저장 논리적 구조



- http 요청 (http 프로토콜 단체 약속 규칙)

폼 - get | 폼입력- post(암호, 게시물내용)

시작라인-제한	uri?요청파라미터들 http 버전	get/post	request.getMethod() request.getRequestURI()
---------	------------------------	----------	--

http 요청 header	브라우저이름, 요청ip,	request.getHeader("user-agent") request.getHeader("....") request.getRemoteAddr()
http 요청 body	post 요청파라미터들 파일업로드-post multipart/form-data	request.getParameter() request.getParameterValues()

- 웹 클라이언트가 요청을 보낼 때, **http 프로토콜**의 **바디** 부분에 데이터를 여러 부분으로 나눠서 전송 - **파일 업로드 - http 요청 body - post 전송**
==> **multipart**

javax.servlet.http.HttpServletRequest 클래스 타입 객체
파일 업로드 지원 메소드가 없다

- 사용자는 javax.servlet.http.HttpServletRequest 의 body 부분을 읽어들이는 입력스트림을 이용해서 Multipart 부분을 잘 나누어서 사용해야 하는데 직접 구현하지 않고 라이브러리를 사용한다.

파일업로드-> 파일명, 파일경로(클라이언트 저장 경로- 보안상 서버 전송 x) , 길이, 내용, +.....

클라이언트	서버(spring mvc)
<pre><form action="a" method=post enctype="multipart/form-data"> <input type=file name="upload" > <input type=text name="id" ></pre>	<pre>pom.xml commons-io.xxxx.jar commons-fileuploadxxx.jar 추가 src/main/webapp/WEB-INF/spring/appServlet servlet-context.xml <beans:bean id="" class="xxxxxMultipartFileResolver" /> @PostMapping("/a") public ? m1(MultipartFile upload, String id){ } xxxxDTO</pre>

클라이언트 파일 업로드 http://localhost:8080/test/fileupload	서버 전달 c:\wupload 경로로파일저장 (파일내용+파일명) 1. 클라이언트 원래파일명 2. 클라이언트 가공 원래파일(UUID값).확장자
내가 서버 업로드파일 다운로드 http://localhost:8080/test/filedownload?파일명=c:\wupload\Wxx	test 컨텍스트는 first dynamic web project 의미하므로 http://localhost:8080/test/xxxx ==> 처리 파일들은 first 프로젝트 내부 파일 이어야 한다

- 파일 다운로드

1. 서버 특정 디렉토리 파일들은 클라이언트 브라우저에 노출되지 않는다
2. 접근할 수 없다
3. 다운로드 구현해야 한다=c=m=v
4. 클라이언트 파일입력<--- 서버파일출력- OutputStream 구현한다
5. 응답 설정 사항

//response.setContentType("text/html;charset=utf-8);--> html 문서 브라우저 실행

response.setContentType("application/download");

response.setHeader("Content-Disposition", "attachment;filename=다운로드파일명");

6. pom.xml , servlet-context.xml 파일 추가 설정 없다

29장

REST API

Representational State Transfer PROTOCOL

HTTP

1> 동기적 SYNCHRONOUS 기본

클라이언트 요청1 – 파일 찾아서 + 서버 자바 실행 결과1 – 응답1-

클라이언트 요청2 – 파일 찾아서 + 서버 자바 실행 결과2 – 응답2

요청3 ---응답3-화면 결과 출력

ASYNCHRONOUS 처리 – 속도빠르다

요청1 – 요청2 - 응답2..응답1

loginform.jsp – 로그인폼 + 이미지 기사 사진 변동없다

loginresult.jsp - 로그인결과출력 + 이미지 기사 사진 변동없다

1>코드 반복 심하다

2> 2개 페이지를 서버로부터 새로 로딩

==>

login.jsp ==> <div>서버로부터응답여부따라서 화면 변동</div>+ 이미지 기사 사진 변동없다
페이지 전체 새로 로딩되는 아니고 페이지 일부만 변경 적용

multi page application ==> MPA

single page application ==> SPA

AJAX – ASYNCHRONOUS JAVA SCRIPT AND XML

1> HTTP 프로토콜 기본 SYNCHRONOUS 방식

2> HTTP 프로토콜 ASYNCHRONOUS 방식 제공하지 않는다

3> 비동기 요청 JAVA SCRIPT 언어 ASYNCHRONOUS 구현 --> JQUERY 1개 함수 구현

4> 비동기 응답 SERVLET+JSP / SPRING MVC

요청1 --- 요청2 ---- 응답2---응답1—화면 구성 – 1개의 페이지 내부 일부 화면 구성변경

원글	
댓글 10개	
더보기 버튼 클릭하면	
댓글 10개 더 추가	

http요청

요청 요청했던 뷰 없어지고 jsp 파일 뷰 출력	html 코드 jsp <h1><%=jsp변수%></h1>	응답
----------------------------------	---------------------------------------	----

ajax

요청 요청했던 뷰 그대로	html-jsp(x) html 태그내부 표현결과 규칙 응답xml => json java script object notation ajax asynchronous java script and xml	응답서버 "<xml> <result> <process> 정상로그인 </process> </result>" "{변수명:값 , , , }"
------------------	---	--

요청 클라이언트 html태그만으로는 불가능 \$.ajax({ 자바스크립트객체 response.logintime }); \$.함수(); \$.함수(function() { }); \$.함수(a=1, b="js" , c=true); var d = { a : 1, b : "js", c: true };	응답 서버 리턴 String --> 뷰이름 리턴 String 뷰이름이 아니라 JSON 형태로 리턴 return "{W"변수명W" : W"값W" , , , }" { "process" : "정상...값" , "logintime" : "2022-08-22" } @ResponseBody @RequestMapping("/...") String m1(){ return "{W"aW" : W"값W" , , , }"; "a:값" 리턴값 }
---	--

<pre>\$.함수({ a : 1, b : "js", c: true }); \$.ajax({변수명 :값 , , ,}); \$.ajax({ url : 'loginajax', data : {id:\$("#id").val(),'password':\$("#password").val()}, type : 'post', //////////서버로 요청 완료////////// //////////서버로부터 응답 시작////////// dataType: 'json', //서버로부터 응답형태 success : function(server){ //response = {"process": "정상로그인사용자"} alert(server.process); }//success });//ajax</pre>	
<pre>{var i =10; i = 11; } i = 12; { let j = 10; j = 11; } j = 12;==>error const k = 10; k = 11;==>error var i = 10; var j = 3.14; var k = "aaa"; var k2 = 'aaaa'; var ar = [1,2,'aaa'];</pre>	

```
var f = function(){ alert(); }
```

```
var obj= {
```

```
  id : "ajax" ,
```

```
  password : 1234,
```

```
  print : function(){ xxxx; }
```

```
};
```

==> 자바 스크립트 객체

json=자바스크립트 객체

```
spring mvc="{}
```

```
{
```

```
  "id" : "ajax" ,
```

```
  "password" : "1234",
```

```
};
```

==> 자바 스크립트 객체 내부 변수명 ""

함수 제외

==> JAVA SCRIPT OBJECT NATATION

==> JSON