**프로젝트 구성요소**

* **DTO3개(BookDto,CustomerDto,FileboardDto), DAO 3개(BookDao,CustomerDao,FileboardDao)는 src폴더에 아래의 폴더는 WebContent폴더에**
* **img 폴더 - 이 프로그램에서 필요한 이미지들(로고, hot.gif, level.gif, re.gif, fileup.jpg)**
* **main 폴더**

**- footer.jsp**

* **header.jsp**
* 로그인 전/ 후에 따른 보여지는 화면이 다르므로 <li> 태그 에 스크립트릿을 열어 if문으로 session을 get해준다

텍스트이(가) 표시된 사진

자동 생성된 설명

* **main.jsp(첫화면)**

header와 footer가 항상 보여지므로 위 아래에 include 해준다

페이징 처리가 되야 하므로 **기본** **세팅을** 해준다

----------------기본 세팅 ------------------

텍스트이(가) 표시된 사진

자동 생성된 설명

PAGESIZE : 페이지당 보여지는 게시글 개수

BLOCKSIZE : 하단 부에 보이는 페이지 수

startRow : 시작번호 = (현재 페이지-1)\* 페이지당 개수 + 1

endRow : 끝 번호 = 시작번호 + 페이지당 개수 -1

CustomerDao cDao = CustomerDao.getInstance(); //싱글톤 이용

ArrayList<CustomerDto> customers = cDao.listCustomer(startRow, endRow) //시작 번호와 끝 번호를 매개변수로 줘야한다.

텍스트, 스크린샷, 은색이(가) 표시된 사진

자동 생성된 설명

이 로직 다시 물어보고 이해하기

startPage는 하단 부에 보여지는 시작하는 페이지

11~20이 블록 사이즈로 10개만 보일 때 11이 startPage

endPage 로직이 헷갈리므로 여러 번 복기.

블록사이즈가 10이고 전체 21페이지가 있다고 가정 했을 때

밑에

1 2 3 4 5 6 7 8 9 10 <다음> -------1

<이전>11 12 13 14 15 16 17 18 19 20 <다음> -----2

<이전> 21 -----3

이렇게 되면 ,

2에서 현재 내가 13페이지 일 때. 시작 페이지는 11페이지다 .

11 + 10 – 1 인 20이 끝 페이지

내가 헷갈린 이유는 endPage라 해서 21페이지인 줄 알았던거 같다.

If endPage > pageCnt 로직부터 이해가 안간다.

->**답변**

EX) 페이지 사이즈 10 , 블럭 사이즈 10

페이지 사이즈 10의 의미 : 1페이지당 나오는 데이터의 수 (한페이지에 나오게 하는 고객 리스트를 10명만 나오게 하겠다)

블록 사이즈 10의 의미 : 아래 페이지 숫자가 보이게 할 때 몇 개까지 보이게 할 것인가 ( 블록 사이즈 10을 잡으면 아래에 나오는 단위로 페이지 넘어가는 숫자가 보임)

페이지가 01 02 03 04 05 06 07 08 09 10  
 11 12 13 14 15 16 17 18 19 20

21 22 23 24 25 26 27 28 29 30

여기서 블록은 3블록이 존재 하는 것 (블록사이즈 10짜리 블록이 3개 )

\*startPage : 01, 11, 21 (각 블록당 존재하는 첫번째 시작 페이지)

\*endPage : 10, 20, 30 (각 블록당 존재하는 마지막 페이지)

\*pageCnt : 페이지의 총 개수 (위에서 페이지의 총갯수는 30개)

endPage 의 숫자가 pageCnt의 숫자보다 적다면? 아직 나올 페이지가 남았다는 뜻

하지만 우린 블록사이즈를 10으로 잡아놨기 때문에, 남은 페이지를 표시하기 위해선 다음버튼을 만들어줘야함

즉 if(endPage < pageCnt)를 위 예시에 적용 시키면

-- endPage (10) < pageCnt(30) 나올게 20개는 더 남음 그러므로 다음 버튼을 만들어줘야해

-- endPage(20) < pageCnt(30) 나올게 10개는 더 남음 그러므로 다음 버튼을 만들어줘야해

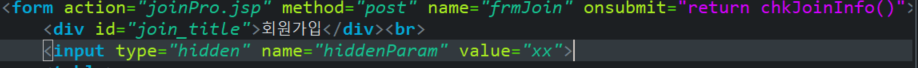
-- endPage(30) = pageCnt(30) endPage숫자와 pageCnt숫자가 동일 그이상 나올 페이지 개수가 남아있지 않기 때문에 다음버튼을 만들어줄 필요가 없음

* **customer폴더**

**- joinForm.jsp**

회원가입 form 만들기 input 노가다 자바스크립트로 가입시 필요한 유효성 검증을 위해

form 태그에 onsubmit 설정



Hidden을 input해주는 이유는?

**답변**

우리가 어떠한 값을 넘겨줄 때 버튼을 통해 넘겨줄 경우엔 onclick = location.href = ‘join.jsp?pageNum=10’ 이런식으로 넘김 .. 하지만 form 태그로 이루어져 있는 jsp 같은 경우엔 다른 jsp에서 넘겨준 파라미터를 받고 넘겨줄 방법이 없음

따라서 값은 넘겨주되, 고객의 눈엔 보이지 않게 넘기기 위해 hidden 타입을 써서 넘겨주는 것 . 내가 구현하고자 하는 jsp가 ㅇㅇForm.jsp 인지 ㅇㅇPro.jsp 인지에 따라 파라미터를 넘겨주는 방식이 좀 다를수 있음

* + - * + **joinPro.jsp**

액션단 이므로 인코딩처리 – 빈 이용 - property이용

dto의 변수들 get 해주기

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - * + **loginForm.jsp**

뷰단이므로 dive에 id줘서 css 적용 시키고 form태그 post방식으로 만들고 input 노가다

가입할 때 만들어 둔 cid session을 get 해주기

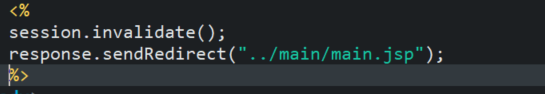
요구사항에 맞게 required 해주기

* + - * + **loginPro.jsp**

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - * + **logout.jsp**



몇 초 후 페이지 이동 시킬 수 있음

|  |
| --- |
| <script> |
|  | setTimeout(function(){ |
|  | location.href = 'main.jsp'; |
|  | }, 2000); |
|  | </script> |

* + - * + **modifyForm.jsp**

CustomerDto customer = (CustomerDto)session.getAttribute("customer");

**CustomerDto 전체를 세션으로 넣어서 이용 할 수 있다 customer세션을 못 받아온다면, 로그인이 안된 뜻이다**

Dto 객체 get 해주기

뷰단이므로 dive에 id줘서 css 적용 시키고 form태그 post방식으로 만들고 input 노가다

label for 이용

* + - * + **modifyPro.jsp**

액션단 이므로 인코딩처리 – 빈 이용 - property이용

텍스트이(가) 표시된 사진

자동 생성된 설명

* **book폴더**

**- bookList.jsp**

// paging 기본 세팅해주기

* + - * **bookRegisterForm.jsp**

보여지는 뷰단 이므로 form태그 안에 input 넣어주는 노동행위

post방식으로 해야되고 enctype을 *"multipart/form-data"*으로 해줘야 한다.

* + - * **bookRegisterPro.jsp**

한글 인코딩처리 - multipartRequest객체 생성 - 첨부된 파일명 get - 그외 파라미터에서 입력된 정보 get – insert

텍스트이(가) 표시된 사진

자동 생성된 설명

객체들의 집합체(Enumeration)을 만들어서 mutipartRequset객체의 파일이름들에 넣어준다

while(Enumeration.hasMoreElement()) paramName에 요소들이 있으면(true) while문 진행

E nextElement() 파라미터 이름이 존재하면 그 요소로 이동시키고, 가리키고 있는 객체를 꺼내 반환한다

image라는 저장소에 request객체의 파일들 이름을 받아 저장한다

// 서버의 bookImg폴더 파일을 소스의 bookImg폴더로 복사

InputStream과 OutputStream을 이용해서 복사+출력을 한다

텍스트이(가) 표시된 사진

자동 생성된 설명

여기서 byte가 하는 일 물어보기

byte[] bs = new byte[(int)serverFile.length()]; //질문!!

Is 로 서버 파일을 받고 ,

os로 저장될 파일 폴더로 복사하며, byte로 저장될 파일의 size(? length대신 size는 안되나? )를 읽는다

while문 질문하기! while문 속 is.read(bs)

// 파일첨부외 파라미터에서 입력된 정보 get

// DB에 insert

**답변**

Byte는 파일을 컴퓨터가 읽을 때 몇바이트식 읽을 것인지를 정해주는 로직이라고 생각하면 됨. serverFile에 넣어두는 파일의 길이 만큼 한 바이트씩 읽겠다 라는 뜻 !

즉 파일을 읽어내는 단위라고 생각해주면 됨

while문은 위에서 몇바이트씩 읽을 건지 정해줬으니, 그걸 읽어라! 라고 다시 선언해주는 부분이라고 생각해주면 됨

while(true) - 돌아라 파일을 처음부터 끝까지 다 읽어낼때까지

int readCnt = is.read(bs) -- bs는 위에 byte 읽어내는 부분에서 받아온 변수인데, bs의 단위만큼 is.read 읽어내라 라는 뜻 그리고 그 읽어낸 아이들의 길이(?)바이트수(?)를 readCnt라는 변수에 담아라 라는 뜻이고,

만약 readCnt가 -1이라는건 더 이상 읽어낼 파일이 없다는 뜻이니 break해서 빠져나오라는 뜻

그리고나서 읽어낸 아이를 os.write(os에 올려라) (bs, 0 , readCnt) bs의 방식으로 0부터 readCnt (파일의 끝)까지 라는 뜻

* + - * **bookContent.jsp**

bid를 get해줘서 상세보기

할인을 안할 때 <del>태그가 적용되는 것 없애기

텍스트이(가) 표시된 사진

자동 생성된 설명

* **bookImg폴더 - 도서등록시 업로드할 파일들이 저장될 폴더**
* **fileboard폴더**

– **fboardList.jsp**

말 그대로 게시판 모양(list)의 뷰단이다 .

페이징 처리가 되야 하므로 기본 세팅 먼저 해준다.

* **pageNum 변수설정, 형변환, 초기값 PAGESIZE, BLOCKSIZE,**
* **싱글톤 호출** FileboardDao fbDao = FileboardDao.getInstance();
* **Dto받기(**ArrayList<FileboardDto>dtos=fbDao.listBoard(startRow, endRow);

Dto 출력 for문에서 답변글일 경우 들여쓰기 처리를 해줘야 한다

텍스트이(가) 표시된 사진

자동 생성된 설명

그 다음은 div class = “paging” 주고 페이징 처리 하기

* + - * + **fboardContent.jsp**

Content나 Detail에서 아래 로직을 항상 먼저 수행해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위 2 줄이 파라미터 받고

getInstance가 싱글톤 객체를 받아온다

**FileboardDto의 정보들을 dao의 getBoard함수에서 fnum을 매개변수로 받아온다.**

텍스트이(가) 표시된 사진

자동 생성된 설명

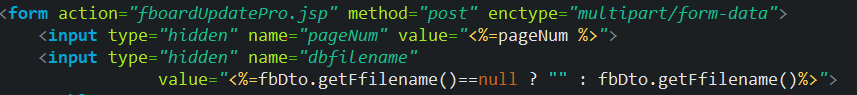
mailto : 에 써놓은 메일에 메일을 보낼 수 있는 페이지로 넘어가게 한다

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - * + **fboardDeleteForm.jsp**
        + **fboardDeletePro.jsp**
        + **fboardUpdateForm.jsp**

content에서 처럼 파라미터 받고, 싱글톤 객체 이용하고, 매개변수로 dto정보 받는다



**기존 로직**

<**input** type=*"hidden"* name=*"dbfilename"*

value=*"*<%=fbDto.getFfilename()%>*"*>

**파일을 첨부하지 않고 수정 할 때, String으로 받아들여서 “null”이 출력되는 버그 발생**

**없애기 위해 value에서 null일 경우와 아닌 경우 나누어서 출력**

* + - * + **fboardUpdatePro.jsp**

실제 저장되는 경로 설정을 해줘야 한다.

String path = request.getRealPath("fileboardUpload");

String ffilename = null; // 첨부파일이 서버에 저장된 이름

MultipartRequest mRequest = null; //객체 초기화

try{

mRequest = new MultipartRequest(request, path, maxSize, "utf-8",new DefaultFileRenamePolicy());

Enumeration<String> params = mRequest.getFileNames();

//bookRegisterPro와 달리 while문 안쓴 이유는?

->>bookRegisterPro에서는 파일첨부를 2개를 해서 while문을 수행 하지만 여기서는 파일첨부를 하나만 하니까 while문을 할 이유가없다.

String param = params.nextElement(); //

ffilename = mRequest.getFilesystemName(param);

이후 파라미터 정보들 get해주기

* + - * + **fboardWriteForm.jsp**

fboardWriteForm.jsp => 원글쓰기

: fsubject, fcontent, ffilename, fpw

fboardWriteForm.jsp?fnum=9&pageNum=1 : 답변글쓰기

: fsubject, fcontent, ffilename, fpw (사용자)

* + - * + : fref, fre\_step, fre\_level(원글)

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - * + **fboardWritePro.jsp**
* **fileboardUpload 폴더- 파일첨부게시판에서 첨부한 파일들이 저장될 폴더**
* **그 외 css**