2024-02-29

- 회원 로그인/로그아웃과 파일업로드 구현(~149)

로그인 : BCryptPasswordEncoder 객체 이용

파일업로드 : CommonsMultipartResolver 이용

다음주 월, 화 회원가입 -> 유효성 검사 패턴 이전보다 더 나간다. / 구분하는 정규화표현식

\*\*\*\*\*\*\*\*\*\*\*\*(p119)회원 로그인/로그아웃과 파일업로드 구현\*\*\*\*\*\*\*\*\*\*\*\*\*\*

~~회원제기법~~

PasswordEncoder를 bean에 저장해서 가지고 와서 사용할거다

회원인코딩 암호화하는 것이다

**--- servlet-context.xml(p121)**

view-controller를 추가하여 /loginForm의 경로는 뷰의 loginForm을 지정한다

loginForm.jsp와 css > member.css를 완성된 소스를 가지고 온다

modal 페이지 아래에 none으로 숨겨놓는다 그리고 필요할 때 띄워주는 것

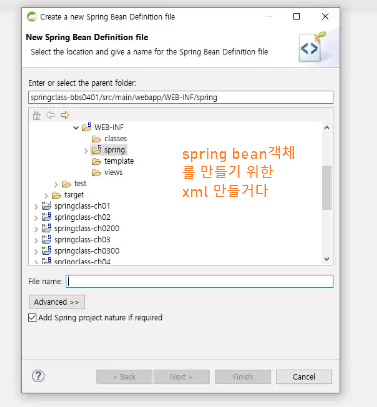
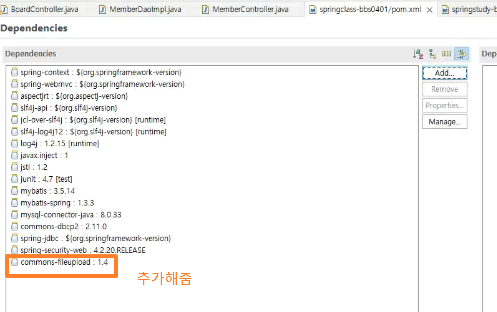
**--- spring-security.xml**

로그인 처리를 위한 Spring-Security 관련 Bean 정의파일 -> 추가한 적이 없어서 우리가 원하는 것이 없다 -> 추가해주기 위해서 pom.xml 에서 Dependencies 에서 spring-secufity 검색

core가 있으면 web이 의존하기 때문에 하나만 등록하면 된다. web등록하면 core도 가지고 온다

bean 등록하는 거니까 bean 태그 이용하면 됨

원래 Spring-Security.xml을 읽는 정의를 root-context.xml에 넣어줘야 하는데 그렇게 하지 않고 web.xml에 <param-value>에 넣어줄거다(root-context와 함께) \_(p123)



\*\*\*\*\* 회원 로그인/로그아웃 구현\*\*\*\*\*\*\*\*\*\*

--- Domain 클래스 -> 회원정보를 저장하는 클래스 = 그릇

**--- MemberDao.java**

게시글 관련된건 BoardDao에서 할것이고 회원관련한것은 MemberDao에서 할거다

Member객체 타입으로 반환한다

-> 하나의 Dao에 계속 추가하는 것이 아니라 Dao도 역할에 따라서 계속 추가해줘야함

**--- MemberMapper.xml**

DB 이름과 모델 클래스 이름이 같아야 읽어올 수 있다

**--- MemberDaoImpl.java**

만든 Mapper namespace를 가지고 온다

@Repository -> 저장소 계층의 component임을 선언한다

한개 가져오는 것 selectOne임

**--- MemberService.java**

회원 정보를 가져오는 getMember를 사용한다 -> 회원정보 왜 가져오냐면 , 회원에 대한 id가 맞는지 안맞는지 확인

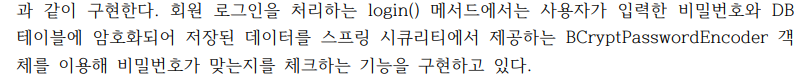
public int login(String id, String pass); ->//로그인 처리 : 로그인 처리되는 것을 확인해야 한다 -> 결과값 뭘로 반환할거냐 boolean으로 반환할 수 있다 그런데 요구상황이 많아서 아이디 없는 경우 비밀번호 틀린것 적어도 3가지 상황이다 (단순히 맞다 틀리다로 할 수 없는 경우) 그때 int로 처리한다 (정수가 제일 좋을 것 같다) -> 세션에 담으려고 함

public Member getMember(String id); -> // 회원 정보를 dao를 통해서 가져오는 메서드

**--- MemberServiceImpl.java**

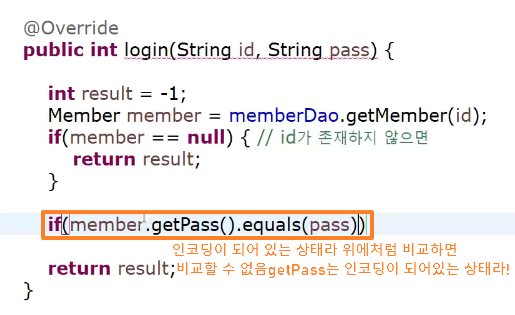
DB관련한 것은 MemberDao 를 의지한다.

비밀번호 암호화 되어 있어서 처리하는데 문제가 생긴다 -> BCryptPasswordEncoder 에 의존한다



Member m = memberDao.getMember(id)

pass는 인코딩이 되어 있는 상태이다. id는 인코딩이 안되어 있는 상황이고



matches는 같은지 확인해주는 것이고 result가 1이면 로그인 성공, 0이면 로그인 실패

// controller에서 필요해서 하나 만든것

@Override

public Member getMember(String id) {

return memberDao.getMember(id);

}

**--- MemberController.java**

// 로그인 폼에서 들어오는 로그인 요청을 처리하는 메서드

login에서 받아야할 것이 1)id, 2)pass 이다.

@PostMapping("/login") -> @RequestMapping 해줘도 되지만 post로 받을 거니까 PostMapping사용해도 된다 -> id와 이름이 다르니까 @RequestParam("userId")으로 지정해준다. pass도 마찬가지 -> 회원정보를 모델에 담기 위해 Model model 도 선언해준다

// 로그인처리

int result = memberService.login(id, pass); -> 회원정보 가지고와서 result에 담아준다

\* if문 쓰기

// 아이디가 존재하지 않으면 - 없다고 알림

// 비밀번호가 다르면 - 다르다고 알림

// 로그인 성공이면 - 세션에 회원정보 저장, 로그인 상태값을 저장

return "redirect: /boardList"; -> 리다이액트로 boardList를 리턴 시킨다.

모델에 담으려면 @Controller 아래 @SessionAttributes("member") -> member와 같은 것은 세션에 담아라

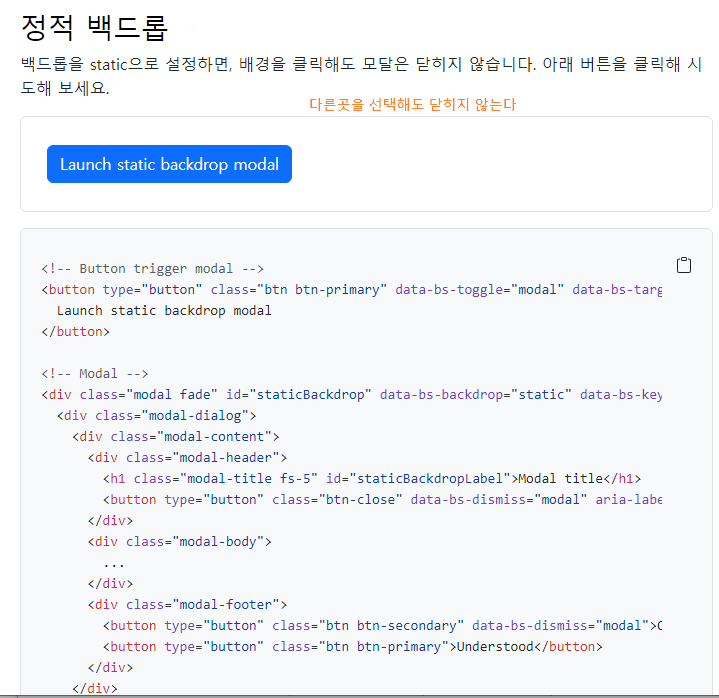
model.addAttribute("member", member) -> 이렇게 해줬으니까 member와 같은 것 세션에 담는 것

**--- index.jsp**

member.css 부분 링크 추가

**--- header.jsp**

'${ sessionScope.isLogin ? "logout" : "loginForm" }' -> 현재 로그인 성공하는 경우를 삼항연산자로 표현하기



**--- MemberController.java**

로그아웃하면 새로운세션 시작하면 됨

\*\*\*\*\*\*\*\*\*\*(p141) 파일 업로드\*\*\*\*\*\*\*\*\*\*\*\*\*

파일업로드 하면 별도 인코딩 방식을 정의해 줘야함

인코딩 타입 기본타임 url 타입이다

MultipartFile 을 이용함, 선택상자 추가해줘야함

**--- WriteForm.jsp**

// 파일 업로드는 포스트(post)방식으로 해야한다

<form name="writeForm" action="writeProcess" id="writeForm"

class="row g-3 border-primary" method="post" enctype="multipart/form-data">

//파일 입력상자 만들어주기

<div class="col-8 offset-md-2">

<label for="file1" class="form-label">파 일</label>

<input type="file" class="form-control" name="file1" id="file1" >

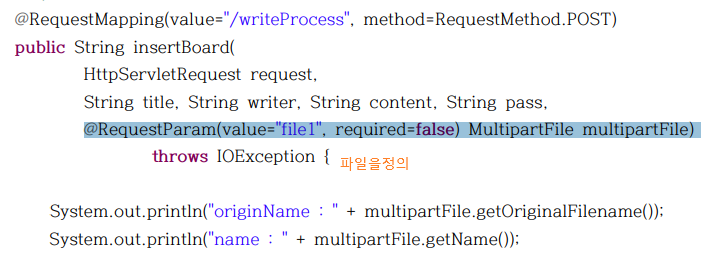
</div>

**--- BoardController.java**

실제 경로를 얻어오기 위함이다

private final static String DEFAULT\_PATH = "/resources/upload/"; -> 업로드한 파일을 저장할 폴더 위치를 상수로 선언

servletContext 파일로 경로를 얻어올 수 있다.

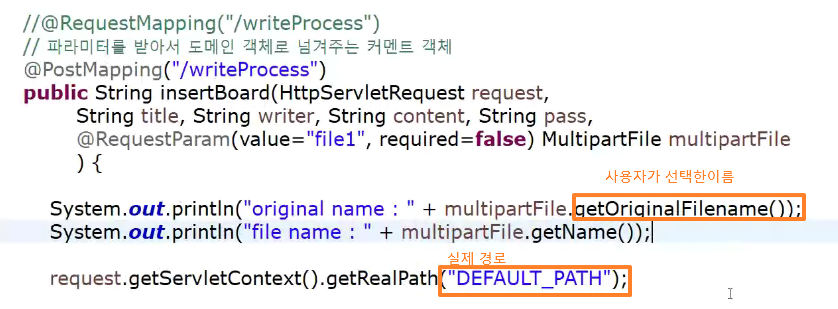


파일업로드 되었는지 확인

if(!multipartFile.isEmpty()) { -> 업로드된 파일 데이터가 존재하면

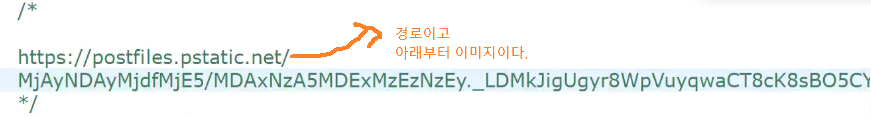
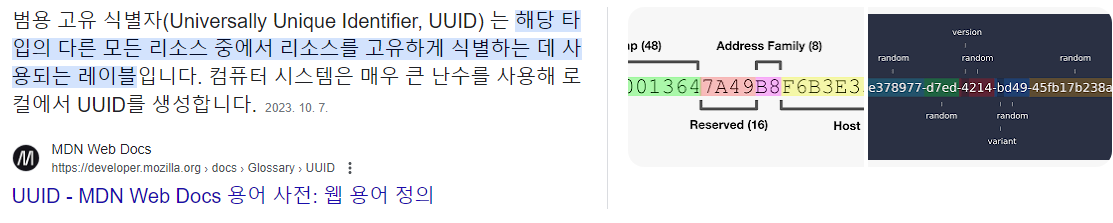
이미지 이름을 재 가공했다는 이야기다

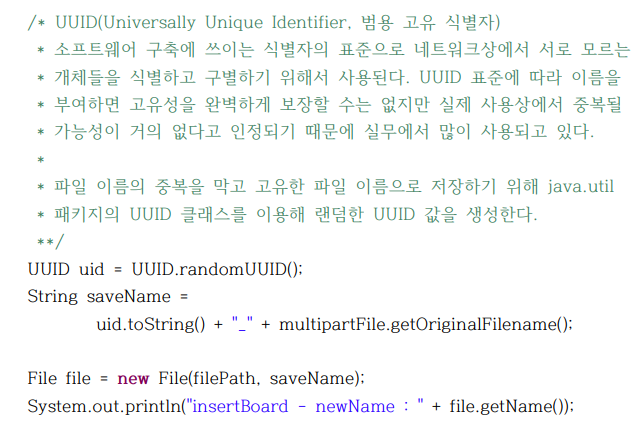
알고리즘의 정책에 의해서 인코딩을 해서 만든 걸 것



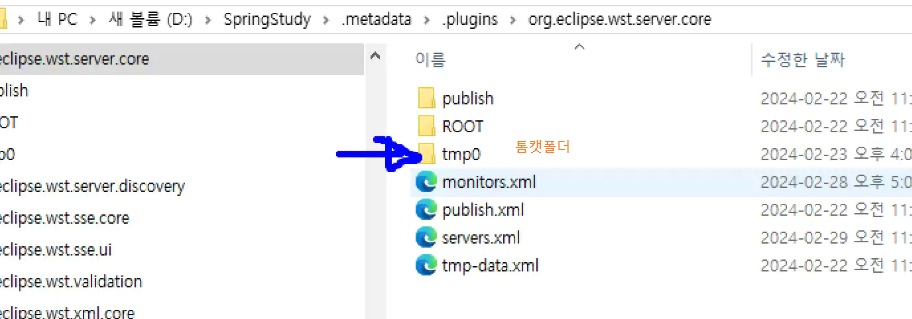
UUID를 통해서 이름에 저장할거다 -> saveName

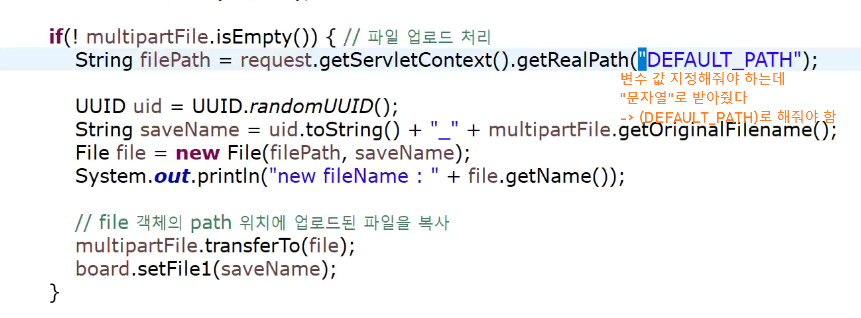
여기 OriginalFilename이 있는데 이럴 경우 한글과 여러가지가 섞여 있다, 그래서UUID를 받는 것

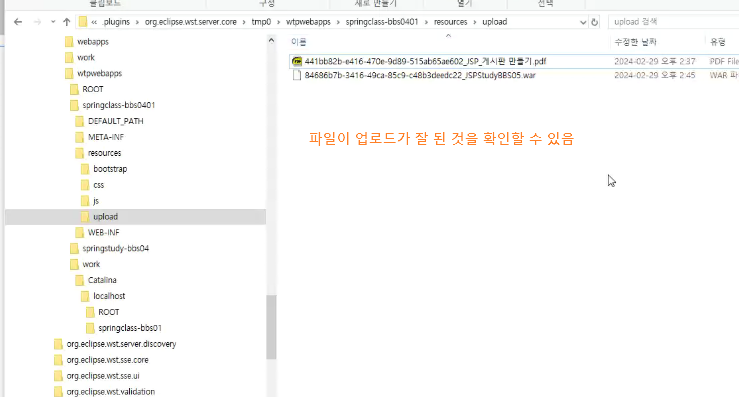
  




File file = new File(filePath, saveName); -> 파일경로와 파일이름을 넣어줘야 파일이 완성이 될 것







-----

데이터 흘러가는 통로 Stream

자바메모리에서 들어오는 것 input Stream, 나가는 것 output Stream.

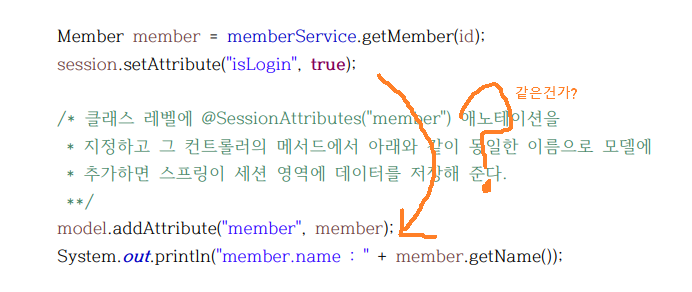
응답객체로 가지고 와야 응답객체로 보낼 것이다. 우리는 output 하는 것

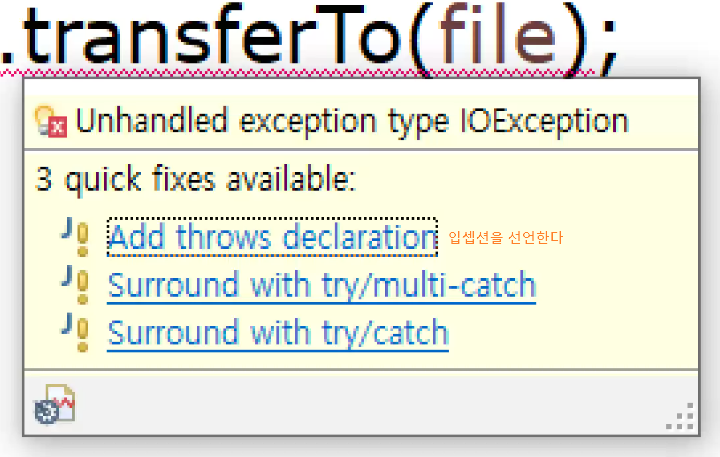
스트림은 바이트 기반이다.

메모리로 읽어와서 써줘야 한다 upload는 바깥에 있는 것이다 input 과 동시에 output을 해줘야 한다

-> FileInputStream을 써준다.

response가 클라이언트한테 응답할 거다





실습

-> 로그인 회원가입

servlet-context를 추가해줄 것

로그인 처리를 위한 bean 추가 -> spring-security.xml

DAO

Mapper

Service

controller

-> view 파일로 감

index, header,