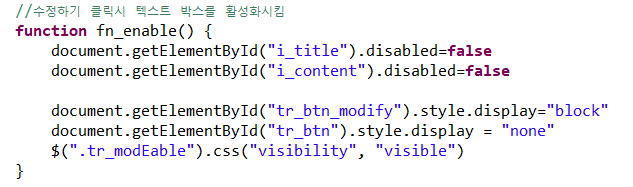
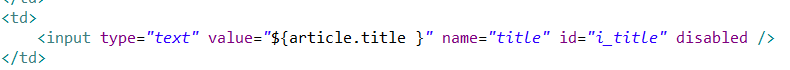
게시판 구현 - 글 수정하기



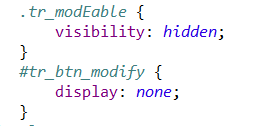
수정하기 클릭시 fn\_enable()

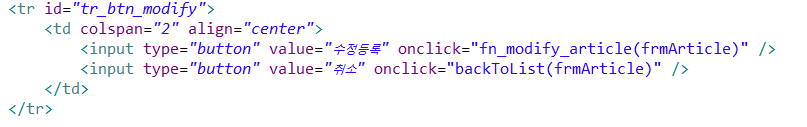


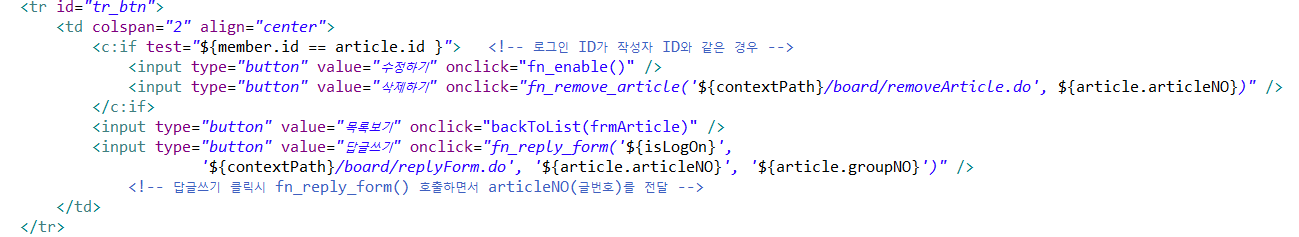




글 제목과 내용이 표시되는 #i\_title과 #i\_content를 입력 가능하게 함







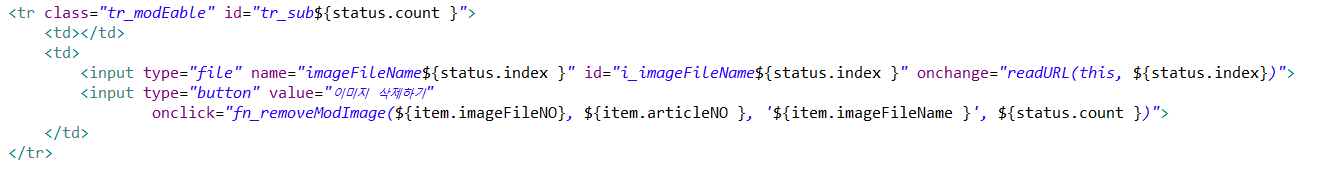
display: none

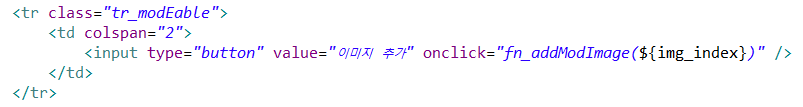
화면 상 어떤 영역도 차지하지 않고 삭제된 것처럼 보임

visibility:hidden

해당 요소가 보이지는 않지만 영역을 차지함

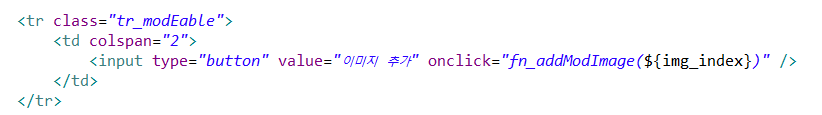
수정등록 버튼을 보이게 하고, 수정하기, 삭제하기 버튼을 가림





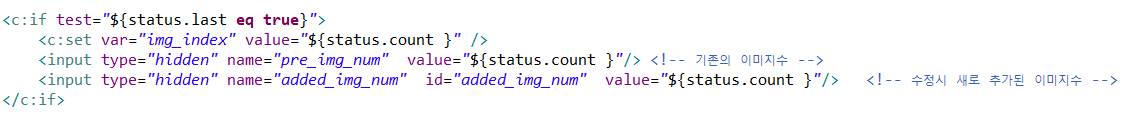
이미지 삭제하기, 이미지 추가하기를 보이게 함

수정하기 - 이미지 추가

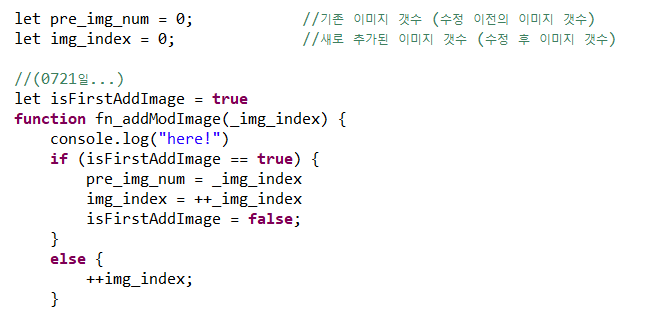


fn\_addModImage는 ${img\_index}를 매개변수로 사용

${img\_index}



img index = 조회된 이미지의 수



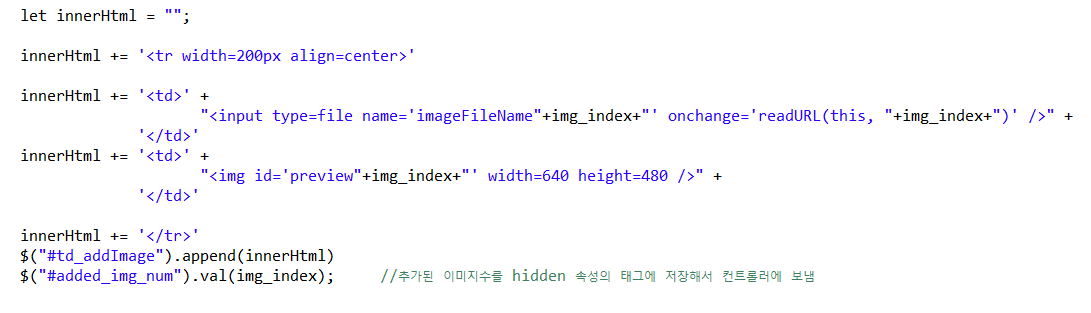
메서드가 처음 호출될 시 변수 isFirstModImage는 true

변수 pre\_img\_num에 매개변수 ${img\_index}값 주입 후, ++\_img\_index

isFirstAddImage = false로 값 변경

isFirstAddImage = false일 경우, 즉 처음 추가하는 이미지가 아닐 경우

++\_img\_index만



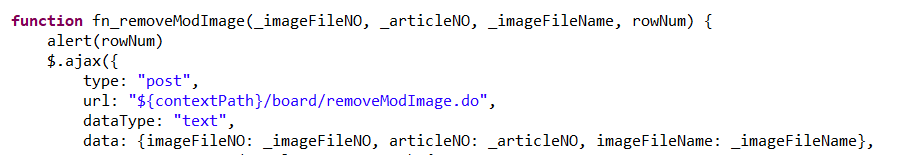
테이블 #td\_addImage에 <input> td와 readURL을 통해 이미지가 표시될

<img> td 추가



added img\_num 값을 수정후 총 이미지 갯수인 img\_index로 변경

수정하기 - 이미지 삭제

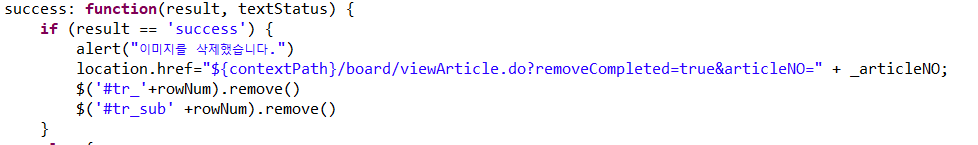


ajax를 통해 imageFileNO, articleNO, imageFileName 값

/board.removeModImage.do에 전송

imageDTO에 imageFileNO와 articleNO를 담아 변수로 service로 전송

해당파일삭제,출력 스트림에 “success” 전송

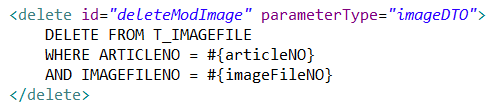


PrintWriter로 전달받은 “success”를 확인

쿼리스트링으로 removeCompleted& 삭제한 articleNO 전송

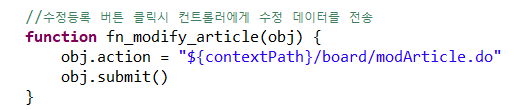
이미지 제목 tr\_${status.count } 삭제

이미지삭제 tr\_sub{status.count } 삭제



수정 등록





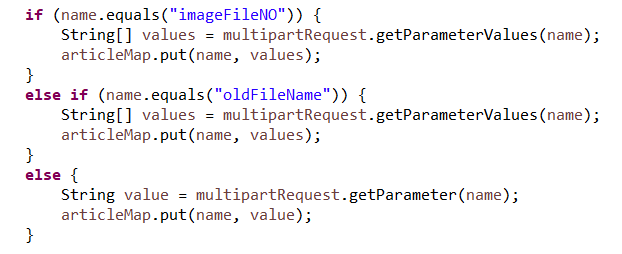
‘수정 등록’ 클릭 시 /board/modArticle.do로 값들 전송

Boardcontroller - /board/modArticle.do



MultipartHttpServletRequest의 속성 getParameterNames()로

view에서 전달받은 name들을 Enumeration enu에 저장



기존 이미지파일 넘버들 imageFileNO의 값list를 articleMap에 저장

기존 이미지파일 이름들 oldFileName의 값list를 articleMap에 저장

나머지 key값의 value들을 묶어 articleMap에 저장



Iterator를 리턴하는 MultipartHttpServletRequest의속성 getFileNames()의 값들을

fileNames에 저장

Iterator fileNames를 순회하면서

MultipartFile의 속성getfile을 통해 parameter의 이름이 fileName인 파일의

정보 구함

MultiPartFIile mFile에 저장

※ MultipartFile 인터페이스는 스프링에서 업로드 한 파일을 표현할 때 사용되는

인터페이스로서, MultipartFile 인터페이스를 이용해서 업로드 한 파일의 이름, 실제 데이터,

파일 크기 등을 구할 수 있다.

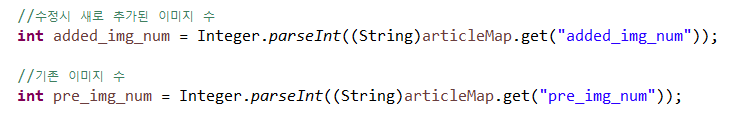
MultipartFile의 속성 getOriginalFileName을 통해 실제 파일의 이름 구함

ARTICLE\_IMAGE\_REPO에 fileName(imageFileName+${satus.inex})이란 파일 생성

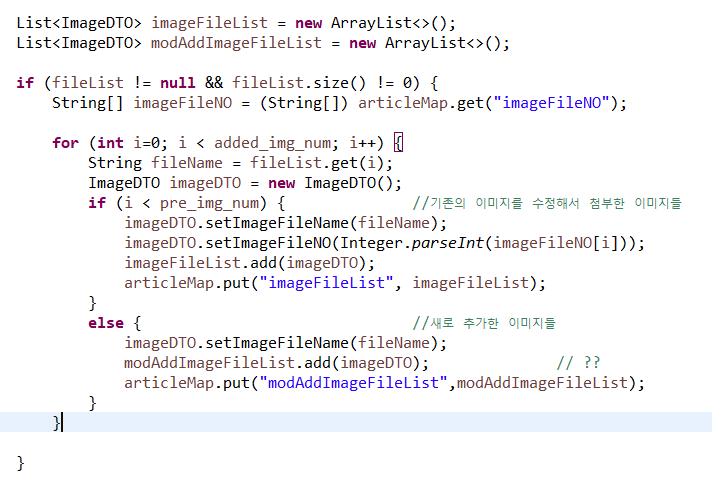
file.getParentFile().mkdirs()를 통해 상위 디렉토리 생성

파일 정보를 담은 MultipartFile mFile을 해당 디렉토리로 이동

fileList = 기존 이미지 + 새 이미지



기존 이미지 수와 수정 후 이미지 수 저장



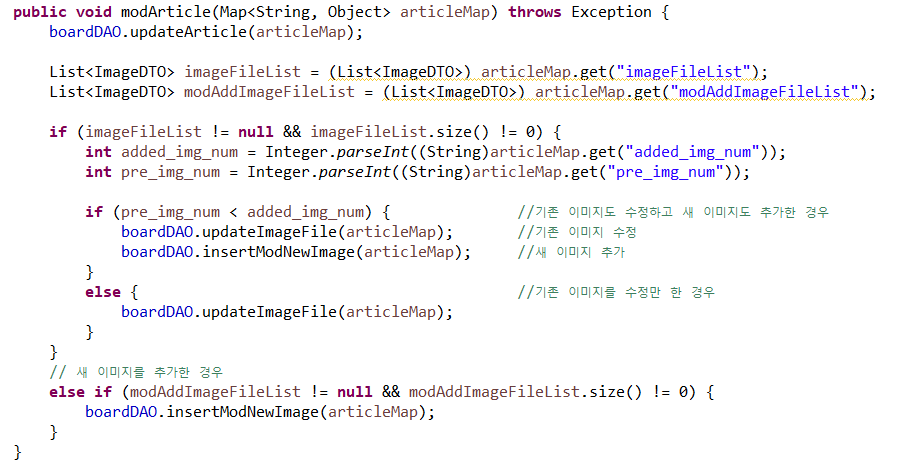
i < pre\_img\_num = 기존 이미지인 경우 : 기존 파일을 수정한 경우

imageDTO fileName저장 후 FileNO를 기존파일번호 i 순으로 로 변경

기존 이미지 파일들을 imageFileList로 articleMap에 저장

새 이미지의 fileName을 imageDTO에 저장 후 modAddImageFileList 저장 후

articleMap에 저장



imageFileList != null && imageFileList.size() != 0 : 이미지 파일을 수정한 경우

 이미지를 추가한 경우



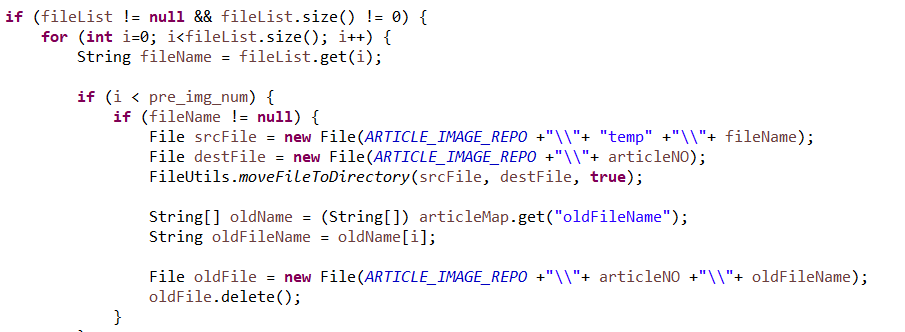
기존 이미지 수정, 새 이미지 추가



이미지만 수정한 경우



새 이미지를 추가한 경우



fileList를 순회하면서

i가 수정전 파일 갯수보다 작다면 => 이전 파일이라면

temp에 저장한 file을 articleNO 폴더로 이동

String 배열 oldName에 articleMap에 저장한 oldFileName 주입

String oldFileName에 i번 째 oldFileName 주입

해당 파일 삭제

