multipart/form-data MIME 타입
> 파트별로 잘라서 리턴해주는 도우미 클래스를 사용해야함.
> 아파치 > 커먼스라이버리

try (**ResultSet rs** = pstmt.getGeneratedKeys()) {
rs.next();
board.setNo(rs.getInt(1));

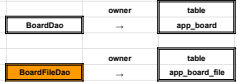여러개의 값을 리턴받기 위해 "ResultSet" 사용
프라이머리 키 조차도 여러개의 레코드/컬럼이 존재할 수 있기때문에 이렇게 사용

*DAO와 테이블

방법1)

BoardDao		owner →		table app_board

★백앙하지 않다면 → owner ↘ table app_board_file
한 개의 DAO가
여러 테이블의 데이터를 관리할 수 있다.

방법2)

BoardDao		owner →		table app_board

BoardFileDao		owner →		table app_board_file

owner(insert/update/delete를 책임진다.)   하나의 테이블의 복수의 DAO를 가질 수 없다.

DAO	→	TABLE

select 은 가능 owner는 만듦 = 조회(select)는 가능하다.

DAO	owner→	TABLE

owner↘

		TABLE

// multipart/form-data 형식으로 요청한 경우
// 요청 데이터의 인코딩을 다음과 같이 지정할 수 없다.
//   request.setCharacterEncoding("UTF-8");

// 멀티파트 데이터 처리하기
// 1) 클라이언트가 보낸 멀티파트 데이터를 임시 보관할 객체 준비
// - 임시 폴더에 저장한다.
DiskFileItemFactory factory = new DiskFileItemFactory();

// 2) 멀티파트 데이터를 분석할 객체 준비
// - 멀티파트 데이터를 파트 별로 쪼개서 이름과 데이터를 분리하여
//   DiskFileItemFactory를 이용하여 임시 폴더에 저장한다.
ServletFileUpload upload = new ServletFileUpload(factory);

// 3) HttpServletRequest 객체를 통해 데이터를 읽어서 멀티파트 데이터를 처리한다.
// - 각각의 파트 정보는 **FileItem** 객체에 담긴다.
// - 그 FileItem 객체들을 List에 담아 리턴한다.
// - 물론 업로드 된 파일의 데이터는 DiskFileItemFactory에서 관리하고 있다.
List<FileItem> items = upload.parseRequest(request);

Servlet API에서 제공하는 multipart/form-data 처리기를 사용하려면
@MultipartConfig(maxFileSize = 1024 * 1024 * 10) //최대 10MB까지 업로드 허용