좋아요 👍 상황을 정리해보면 다음과 같습니다.

* **요구사항**
  + 클라이언트 → 게시글 제목(title), 내용(content), 첨부파일 2개(file1, file2) 전송
  + **DB 테이블 구성**
    - board 테이블: id, title, content, createdAt …
    - board\_attach 테이블: id, board\_id(FK), file\_name
  + **JPA 사용**
  + 파일 자체는 디렉토리에 저장, DB에는 파일명만 저장

아래는 구조를 명확히 한 **Service + Repository** 예시 코드입니다.  
(컨트롤러 단에서 MultipartFile을 받아서 Service로 넘기는 구조를 가정합니다.)

**📁 엔티티(Entity)**

// Board.java

package com.example.demo.entity;

import jakarta.persistence.\*;

import lombok.\*;

import java.util.ArrayList;

import java.util.List;

@Entity

@Getter @Setter

@NoArgsConstructor @AllArgsConstructor @Builder

public class Board {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String content;

@OneToMany(mappedBy = "board", cascade = CascadeType.ALL, orphanRemoval = true)

private List<BoardAttach> attachments = new ArrayList<>();

public void addAttachment(BoardAttach attach) {

attachments.add(attach);

attach.setBoard(this);

}

}

// BoardAttach.java

package com.example.demo.entity;

import jakarta.persistence.\*;

import lombok.\*;

@Entity

@Getter @Setter

@NoArgsConstructor @AllArgsConstructor @Builder

public class BoardAttach {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String fileName;

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "board\_id")

private Board board;

}

**📁 Repository**

// BoardRepository.java

package com.example.demo.repository;

import com.example.demo.entity.Board;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BoardRepository extends JpaRepository<Board, Long> {

}

// BoardAttachRepository.java

package com.example.demo.repository;

import com.example.demo.entity.BoardAttach;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BoardAttachRepository extends JpaRepository<BoardAttach, Long> {

}

**📁 Service**

// BoardService.java

package com.example.demo.service;

import com.example.demo.entity.Board;

import com.example.demo.entity.BoardAttach;

import com.example.demo.repository.BoardRepository;

import lombok.RequiredArgsConstructor;

import org.springframework.stereotype.Service;

import org.springframework.web.multipart.MultipartFile;

import java.io.File;

import java.io.IOException;

import java.nio.file.Path;

import java.nio.file.Paths;

@Service

@RequiredArgsConstructor

public class BoardService {

private final BoardRepository boardRepository;

private final String UPLOAD\_DIR = "C:/upload"; // 실제 저장 경로 수정 필요

public Board saveBoard(String title, String content, MultipartFile[] files) throws IOException {

Board board = Board.builder()

.title(title)

.content(content)

.build();

// 파일 저장 및 BoardAttach 생성

if (files != null) {

for (MultipartFile file : files) {

if (!file.isEmpty()) {

// 파일 실제 저장

String originalName = file.getOriginalFilename();

Path filePath = Paths.get(UPLOAD\_DIR, originalName);

file.transferTo(filePath.toFile());

// 파일명만 DB에 저장

BoardAttach attach = BoardAttach.builder()

.fileName(originalName)

.build();

board.addAttachment(attach);

}

}

}

// JPA cascade 덕분에 board 저장 시 attachments도 함께 저장됨

return boardRepository.save(board);

}

}

**📁 Controller 예시 (참고)**

@PostMapping("/boards")

public ResponseEntity<?> uploadBoard(

@RequestParam String title,

@RequestParam String content,

@RequestParam(required = false) MultipartFile[] files

) throws IOException {

Board saved = boardService.saveBoard(title, content, files);

return ResponseEntity.ok(saved.getId());

}

**✅ 요약**

* Board ↔ BoardAttach 는 **1:N 관계**
* 파일은 실제 디렉토리에 저장
* DB에는 파일명만 저장
* CascadeType.ALL 로 board 저장 시 attach 자동 저장

원하신다면 👉

* **파일명 중복 방지 (UUID 적용)**
* **파일 확장자 제한 (zip, jpg 등)**
* **파일 용량 제한 설정**  
  까지 반영한 버전도 만들어드릴까요?