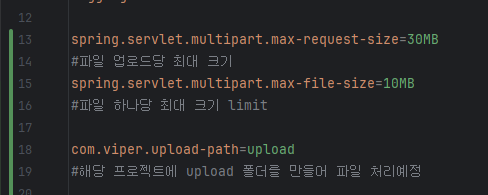
**#파일 업로드**

사실 파일 업로드 관련해서는 별도의 서버를 만들어서 해당 서버가 관리하는 것이 가장 좋다.

이미지 조회와 같은 request들은 굉장히 많이 발생하기 때문에.

보통 db에는 파일 경로만 저장, 보안상 중요한 파일은 db에 직접 업로드 할 수도 있다.

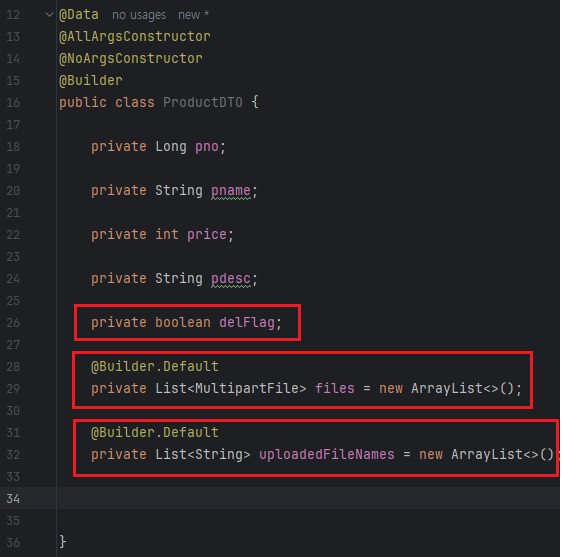
1. **properies파일에 파일 처리 관련 설정 추가**



1. **상품 등록과 조회 시 사용할 DTO 만들기**

* 상품 등록 시 : multipartfile 업로드 되는 파일 처리
* 상품 조회 시 : 경로로 파일 acess





**delFlag**

상품은 실제 삭제가 되는 것이 아닌 삭제가 되는 것처럼 표시해야한다.

외래키로 엮여 있어 상품 삭제 시 통계가 망가질 수 있기 때문에

회원 탈퇴도 마찬가지다. 탈퇴 후 동일 아이디로 재가입시 재가입이 불가능하다.

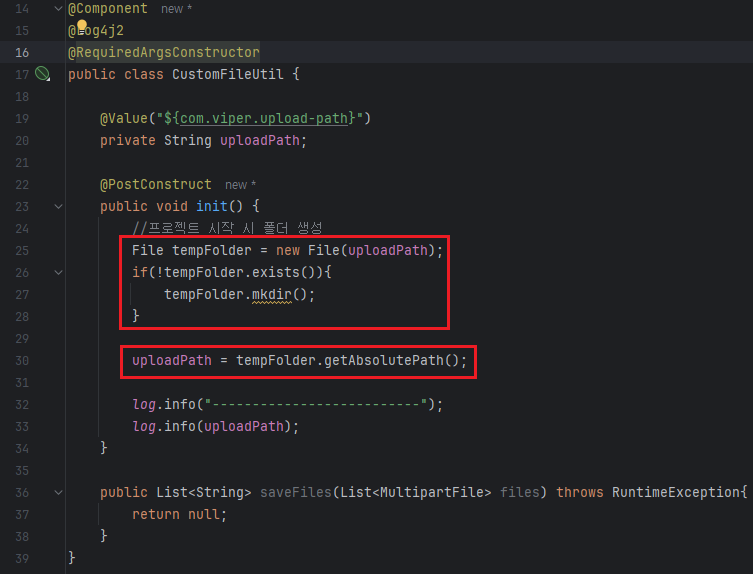
실제로 삭제한 것이 아니기 때문에.

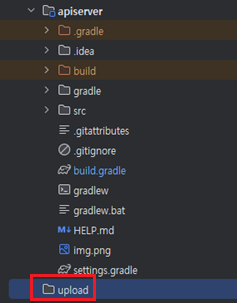
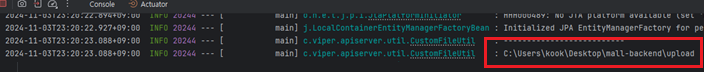
**업로드 실제 파일 리스트와 파일 이름 리스트 객체**

1. **업로드 파일 저장 폴더 처리**

프로젝트 시작 시 해당 경로에 upload 폴더가 없다면 생성하도록 @PostConstruct 활용하여 작성

절대경로상 어떤 위치에 저장 될지 로그 찍어보기





이렇게 실행 시 업로드를 위한 폴더의 생성됨과 절대경로를 로그로 확인해볼 수 있다.

1. **파일 업로드 처리**

동일 이름의 파일 처리를 위해서 UUID와 파일이름을 결합해서 사용

파일 이름과 저장 경로를 결합하여 path 객체를 생성 후에

해당 경로에 파일을 저장한다.

업로드 되었다면 업로드 된 이름을 리스트에 추가하여 반환한다.

