**#페이징 처리**

**페이지 번호 0부터임을 주의**

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

import는 죄다 springframework.domain으로 할 것.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

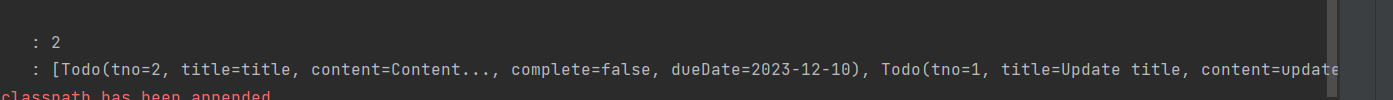
자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명



데이터가 부족하니까 데이터 추가 삽입

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

데이터가 늘어나니까 count 쿼리가 하나 더 날라가는게 보인다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**#QueryDSL**

**쿼리dsl 설정 추가**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Gradle reload를 해주고 나면 프로젝트 build 부분에 아래와 같이 추가 된 것들을 확인할 수 있다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

gradle -> other -> complieJava를 실행해보자

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

실행하고나면 Q클래스(**Q도메인**)가 만들어진다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**여기까지가 스프링부트 3.1버전에서 QueryDSL 세팅하는 방법이다**

**#QueryDSL 사용해보기**

search 패키지를 만들고 TodoSearch인터페이스를 만든다.

**리턴 타입은 일단 Page로 해주기로 한다**

**텍스트, 멀티미디어 소프트웨어, 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**QueryDSL을 사용한다면 인터페이스 이름 뒤에 반드시 Impl로 끝나야한다**

**생성자는 어차피 도메인 클래스를 지정하게 되어있다. 여기서는 Todo 클래스를 넣어주면 되겠다.**

**일단 로그 추가해주고 TodoSearch 오버라이딩 메서드 그대로 두자.**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**레파지토리에 TodoSearch도 상속받도록 추가해준다.**

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

gradle 파일에서 anotation processor 부분에 querydsl 관련 설정 들여쓰기 안맞으면 아래와 같이 오류남.

아래는 맞춰 수정한거. 뭔가 이상한데 왜그런진 잘 모르겠음. 아예 삭제했다가 들여쓰기 맞춰서 다시 reload 하니까 오류 없어졌음.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

본격적으로 쿼리를 작성하고 테스트해보자

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

작성한 서치 메서드를 태스트해보자

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**＃최신 페이징 기술**

**pageable을 그대로 사용할 수 있어서 편리함.**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명