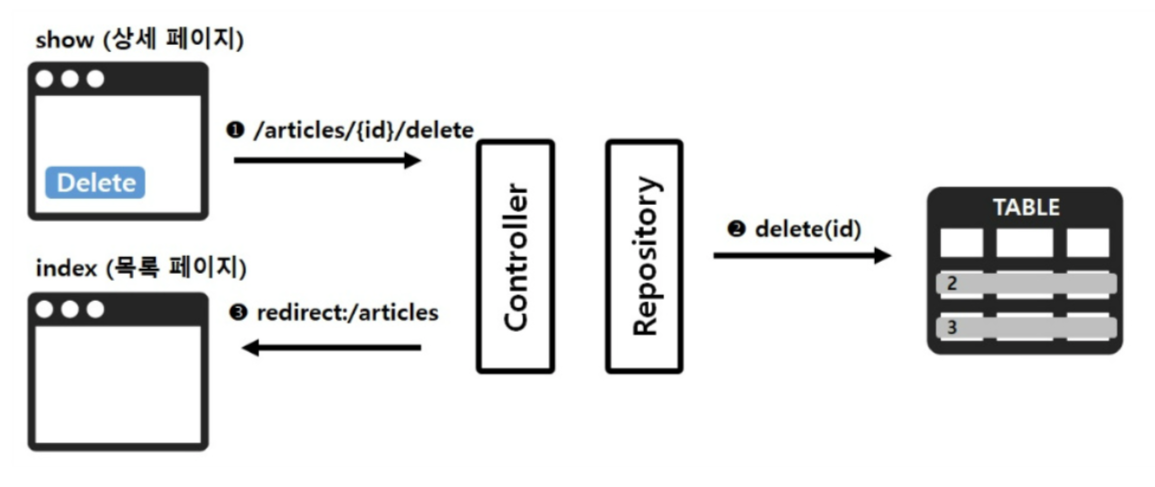
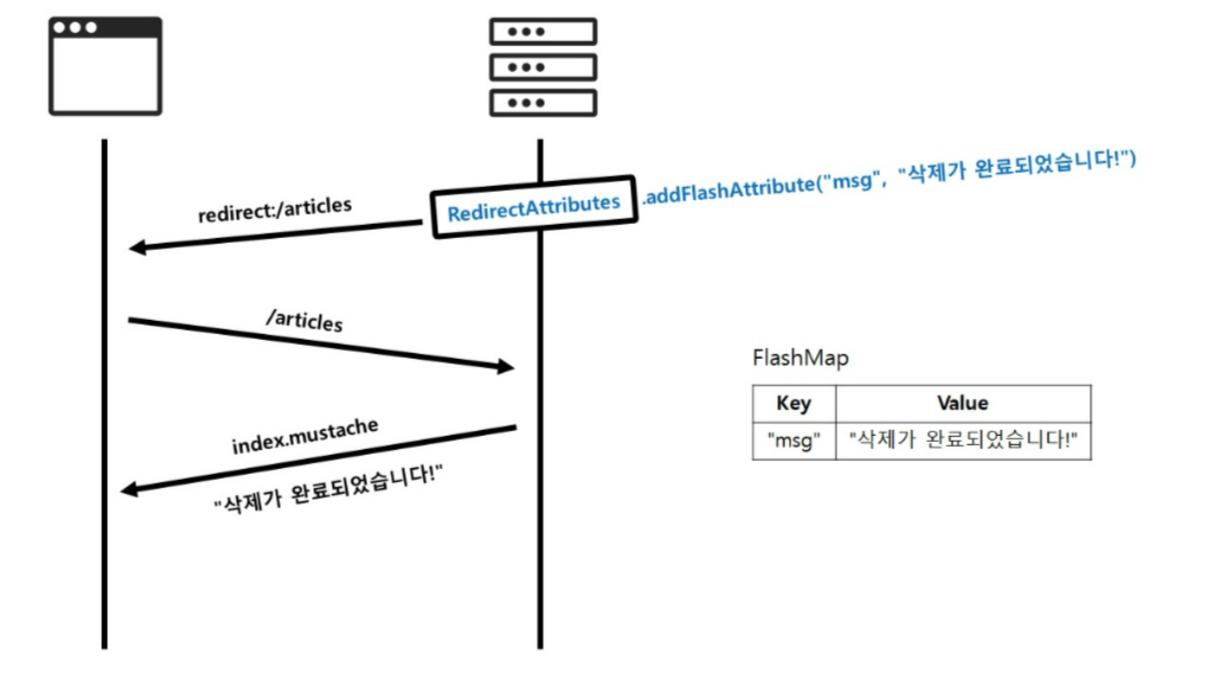
게시글 삭제하기 : DELETE

1. 클라이언트가 HTTP 메소드로 특정 게시글의 삭제를 요청한다
2. 삭제 요청을 받은 컨트롤러는 리파지터리를 통해 DB에 저장된 데이터를 찾아 삭제한다. 이 작업은 기존 데이터가 존재하는 경우만 진행
3. 삭제가 완료되면 클라이언트를 결과 페이지로 리다이렉트한다.



리다이렉트된 페이지에서 일회성으로 사용할 데이터를 등록하여 보내기



Delete 버튼 추가하기

Show.mustache 파일 수정하기

Delete 컨트롤러 생성

get방식을 이용하여 삭제요청을 받아 처리하기

Delete() 메서드 기본 틀 제작하기

* ArticleController에서 delete() 메소드 작성하기

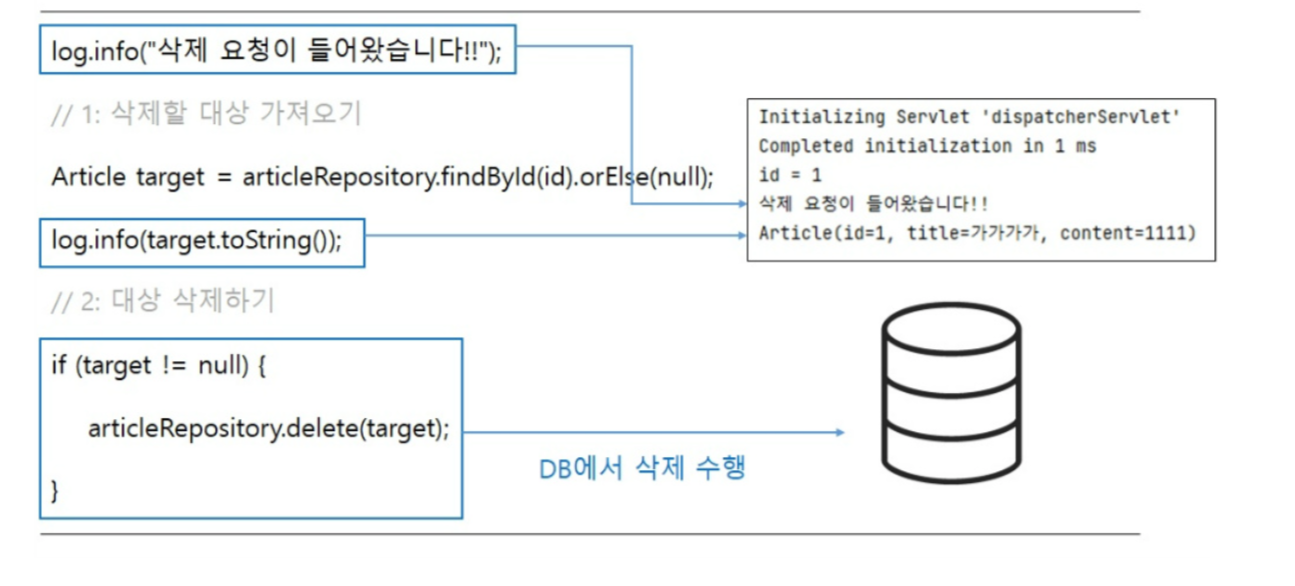
세부분으로 나누어 처리하기

// 삭제할 대상 가져오기

1. DB에 접근하여 데이터를 처리할때는 JPA의 리파지터리를 이용한다

// 대상 엔티티 삭제하기

1. Target에 저장되었다면 삭제를 수행하기



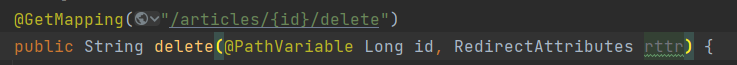
// 결과 페이지로 리다이렉트하기!

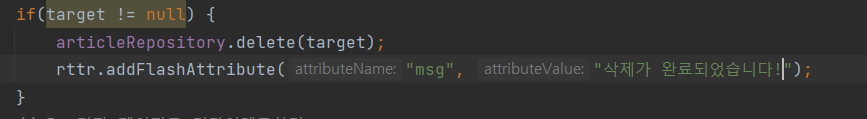
* Delete() 메소드의 return 문에 리다이렉트를 작성하기

“redirect:/articles”;

삭제완료 메시지 남기기

RedirectAttributes 객체를 활용하여 메시지 남기기





삭제 요청 메소드 다음에 바로 붙이기

최종 정리

* 클라이언트가 특정 게시글의 삭제 요청을 하게되면 컨트롤러의 delete() 메소드에서 @GetMapping으로 받기
* Delete 메소드가 삭제 대상을 찾기 위해 대푯값인 id가 필요
* 이를 위해 @PathVariable 어노테이션을 사용. 이는 @GetMapping의 url에서 중괄호에 둘러싸인 값을 매개변수로 가져옴
* Repository는 가져온 id로 DB에서 삭제 대상을 찾고, delete 메소드로 데이터를 삭제 이때, DB 내부에서는 DELETE라는 SQL문이 자동 수행
* 삭제 작업 완료시 결과 페이지로 리다이렉트
* 메시지 출력을 위해 RedirectAttributes 객체의 addFlashAttribute() 메소드를 사용

