데이터 삭제하기



**Delete 버튼 생성**

show.mustache에 delete 버튼 생성

<a href="/articles/{{article.id}}/delete" class="btn btn-success">Delete</a>

원래는 DELETE 요청을 써야하지만  
HTML에서 GET,POST만 사용할 수 있기 때문에  
임의로 GET 요청을 사용하겠다.  
나중에 JS로 DELETE요청을 진행해보자.

**삭제 요청 받기**

ArticleController.java

// @DeleteMapping("/articles/{id}/delete")

@GetMapping("/articles/{id}/delete")

public String delete() {

log.info("삭제 요청");

return null;

}

**삭제 처리 개요**

@GetMapping("/articles/{id}/delete")

public String delete(@PathVariable Long id) {

log.info("삭제 요청");

// 1: 삭제 대상을 가져온다.

Article target = articleRepository.findById(id).orElse(null);

// 2: 대상을 삭제한다.

if(target != null) {

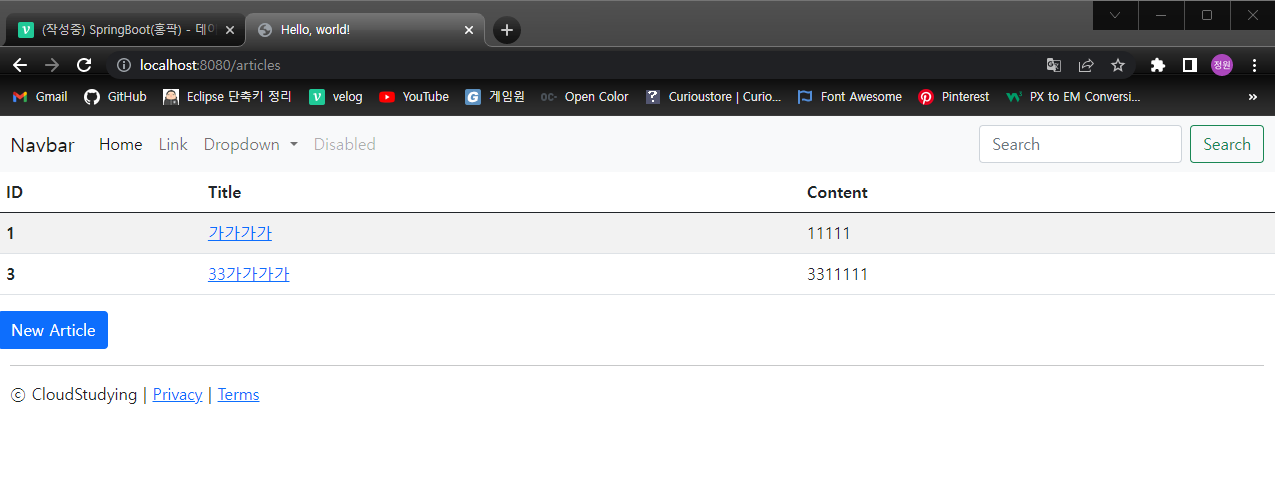
articleRepository.delete(target);

}

// 3: 결과 페이지로 리다이렉트한다.

return "redirect:/articles";

}

삭제완료 👌  


**삭제 메시지 처리**

RedirectAttributes를 이용해서 msg를 전달한다.  
addFlashAttribute는 1회용이다.

@GetMapping("/articles/{id}/delete")

public String delete(@PathVariable Long id, RedirectAttributes rttr) {

log.info("삭제 요청");

// 1: 삭제 대상을 가져온다.

Article target = articleRepository.findById(id).orElse(null);

// 2: 대상을 삭제한다.

if(target != null) {

articleRepository.delete(target);

rttr.addFlashAttribute("msg", "삭제가 완료되었습니다.");

}

// 3: 결과 페이지로 리다이렉트한다.

return "redirect:/articles";

}

redirect:/articles로 msg가 전달되기 때문에  
header에서 msg를 뿌려주자.

<!-- alert msg -->

{{#msg}}

<div class="alert alert-primary alert-dismissible">

{{msg}}

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

</div>

{{/msg}}

삭제 완료 👌👍✨  
