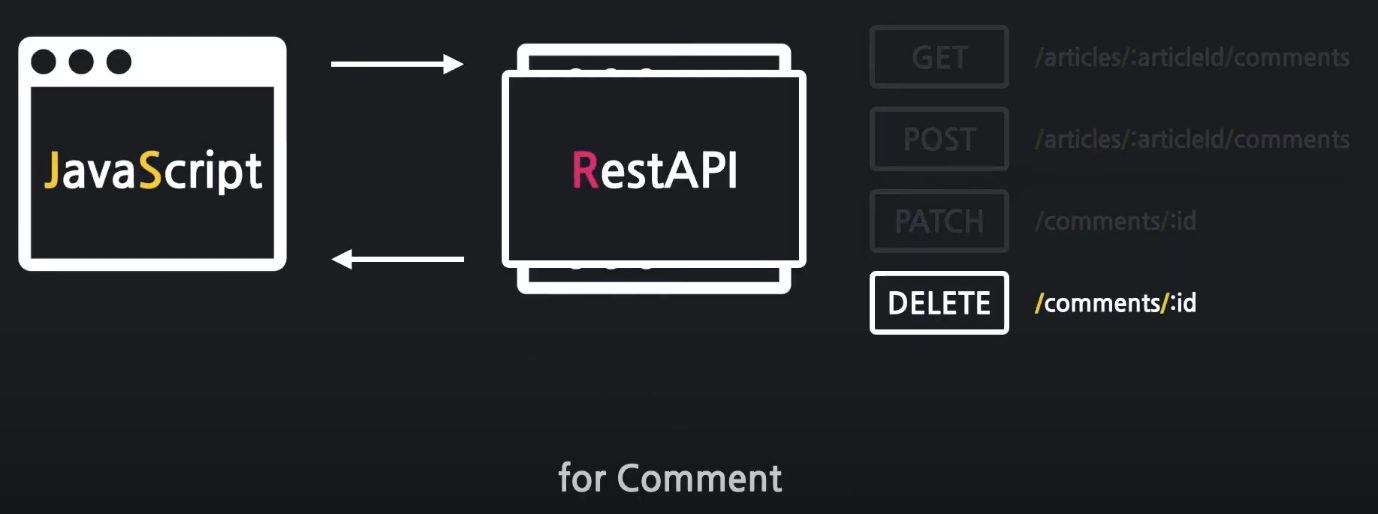
댓글 삭제 with 자바스크립트(JS)





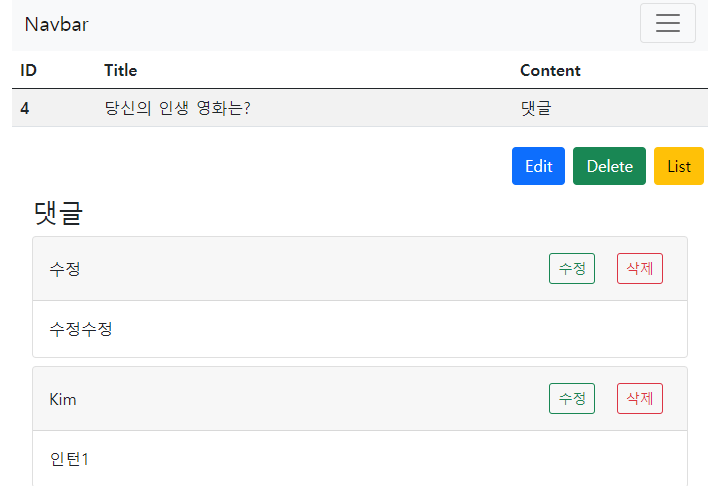
**삭제 버튼 추가**

\_list.mustache에 수정버튼 옆에 삭제 버튼 추가.

<button type="button"

class="btn btn-sm btn-outline-danger m-2 comment-delete-btn"

>삭제</button>



**클릭 이벤트 처리**

<script>

{

// 삭제 버튼 선택

const commentDeleteBtn = document.querySelector(".comment-delete-btn");

// 삭제 버튼 이벤트 처리

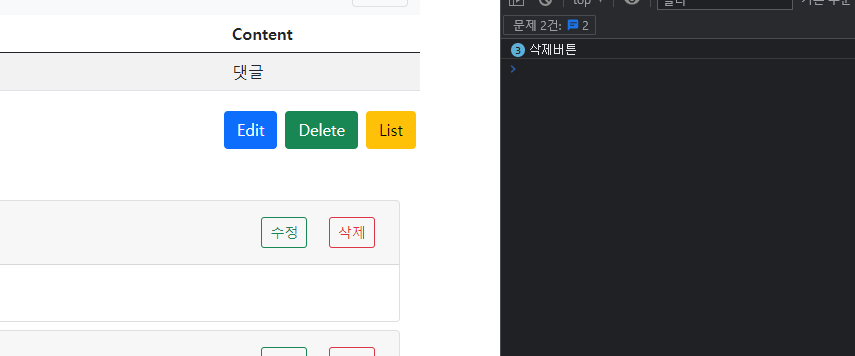
commentDeleteBtn.addEventListener("click", () => {

console.log("삭제버튼");

});

}

</script>



삭제버튼 클릭시 콘솔 출력이 잘 된다.  
❌ 하지만 다른 삭제버튼을 클릭하면 출력이 안된다.  
왜 그럴까??

querySelector는 첫번째 1개의 요소만 선택하기 때문에 다른 삭제버튼을 클릭해도 이벤트가 발생하지 않는것이다.

그렇기 때문에 **querySelectorAll**을 사용하고  
여러게의 버튼을 각각 꺼내서 이벤트를 걸어야하기때문에  
**forEach** 사용.

**forEach**



**댓글 삭제 처리 코드 수정.**

<script>

{

// 삭제 버튼 선택

const commentDeleteBtns = document.querySelectorAll(".comment-delete-btn");

// 삭제 버튼 이벤트 처리

commentDeleteBtns.forEach(btn => {

// 선택한 삭제 버튼 요소들이 하나씩 btn으로 들어오고 각 버튼의 이벤트 처리를 등록

btn.addEventListener("click", () => {

console.log("삭제버튼클릭");

});

});

}

</script>

이제 다른 버튼들도 다 콘솔 출력 확인이 가능해졌다.😍

**삭제 댓글 id 가져오기**

몇번 댓글의 삭제버튼이 클릭 되었는지 확인하기 위해  
id가져오기.

삭제 버튼에 속성으로 댓글의 id를 넣어준다.(데이터추가)  
(commentDtos.id)

<!-- 댓글 삭제 버튼 -->

<button type="button"

class="btn btn-sm btn-outline-danger m-2 comment-delete-btn"

data-comment-id="{{id}}"

>삭제</button>

스크립트 추가

{

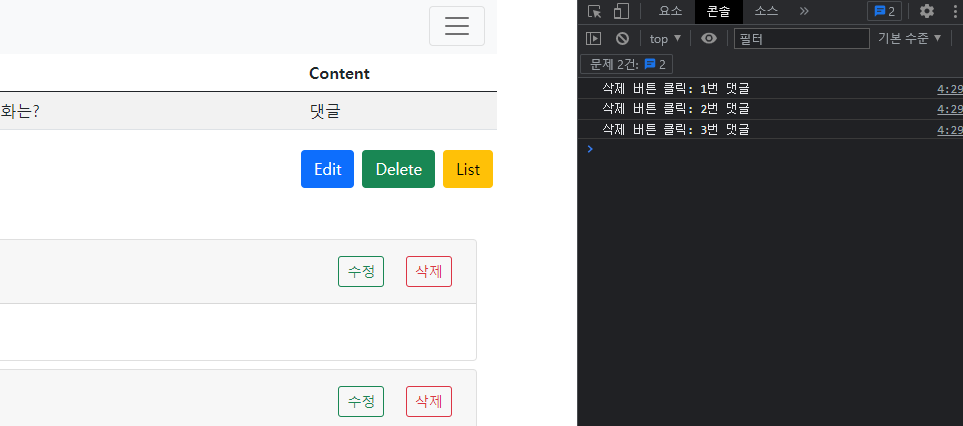
// 삭제 댓글 id 가져오기

// 속성이 data-comment-id인 속성을 거져와서 commentId에 넣기

const commentId = commentDeleteBtn.getAttribute("data-comment-id");

console.log(`삭제 버튼 클릭: ${commentId}번 댓글`);

}

삭제할 대상 id 잘 가져온다. 👌  


**삭제 API 호출 및 처리**

// 삭제 API 호출 및 처리

const url = `/api/comments/${commentId}`;

fetch(url, {

method:"DELETE"

}).then(response => {

// 댓글 삭제 실패 처리

if(!response.ok) {

alert("댓글 삭제 실패!");

return;

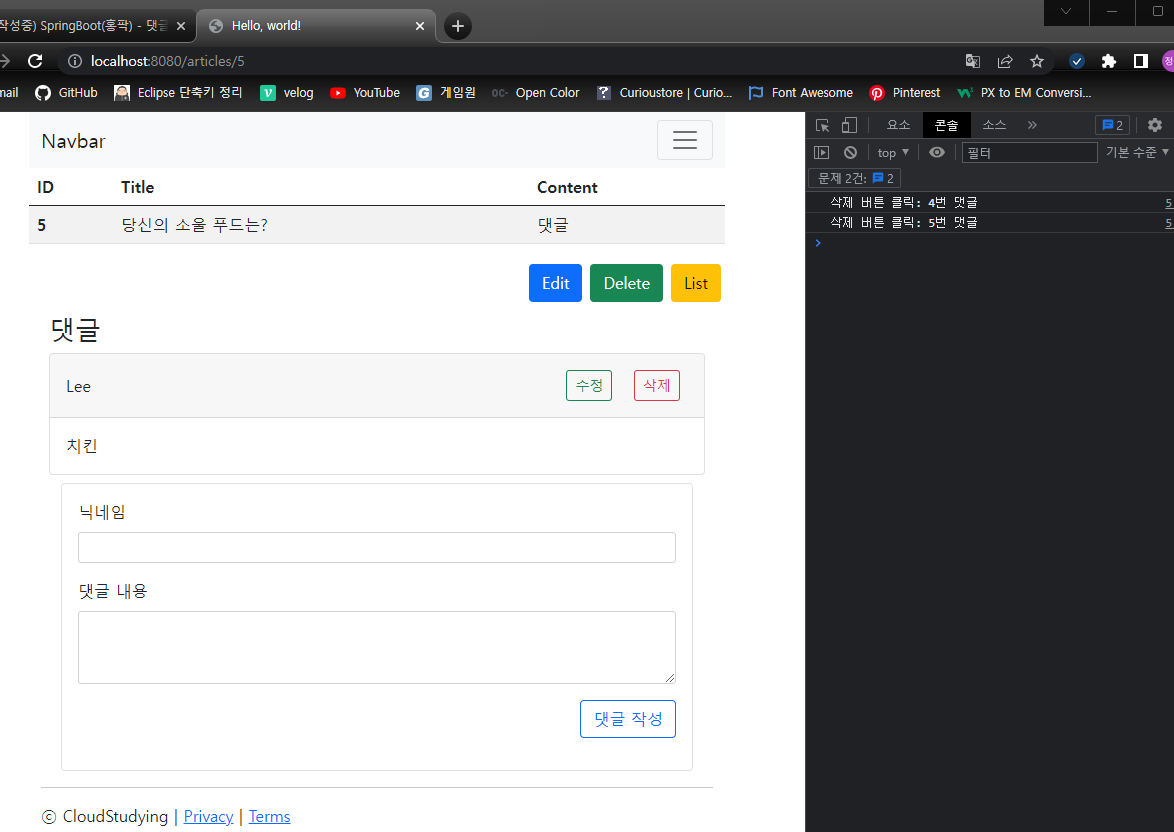
}

// 삭제 성공 시, 댓글을 화면에서 지움

const target = document.querySelector(`#comments-${commentId}`);

target.remove();

});

삭제 버튼 클릭하면 새로고침없이 바로바로 삭제된다.  


**삭제와 새로고침의 차이?**

삭제 처리 후 reload를 하면 새로고침이 된다.

window.location.reload();

**전체코드**

**\_list.mustache**

<div id="comments-list">

{{#commentDtos}}

<div class="card m-2" id="comments-{{id}}">

<div class="card-header d-flex justify-content-between align-items-center">

{{nickname}}

<div>

<!-- Button trigger modal -->

<button type="button"

class="btn btn-sm btn-outline-success m-2 "

data-bs-toggle="modal"

data-bs-target="#comment-edit-modal"

data-bs-id="{{id}}"

data-bs-nickname="{{nickname}}"

data-bs-body="{{body}}"

data-bs-article-id="{{articleId}}"

>수정</button>

<!-- 댓글 삭제 버튼 -->

<button type="button"

class="btn btn-sm btn-outline-danger m-2 comment-delete-btn"

data-comment-id="{{id}}"

>삭제</button>

</div>

</div>

<div class="card-body">

{{body}}

</div>

</div>

{{/commentDtos}}

</div>

<!-- Modal -->

<div class="modal fade" id="comment-edit-modal" tabindex="-1">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">댓글 수정</h5>

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

</div>

<div class="modal-body">

<!-- 댓글 수정 폼-->

<form>

<!-- 닉네임 입력 -->

<div class="mb-3">

<label class="form-label">닉네임</label>

<input type="text" class="form-control form-control-sm" id="edit-comment-nickname">

</div>

<!-- 댓글 본문 입력 -->

<div class="mb-3">

<label class="form-label">댓글 내용</label>

<textarea type="text" class="form-control form-control-sm" rows="3" id="edit-comment-body"></textarea>

</div>

<!-- 히든 인풋 -->

<input type="hidden" id="edit-comment-id">

<input type="hidden" id="edit-comment-article-id">

<!-- 전송 버튼 -->

<div class="d-flex justify-content-end">

<button type="button" class="btn btn-outline-primary btn-m" id="comment-update-btn">댓글 수정 완료</button>

</div>

</form>

</div>

</div>

</div>

</div>

<!-- 모달 이벤트 처리 -->

<script>

{

// 모달 요소 선택

const commentEditModal = document.querySelector("#comment-edit-modal");

// 모달 이벤트 감지

commentEditModal.addEventListener("show.bs.modal", function(e) {

// 트리거 버튼 선택

const triggerBtn = e.relatedTarget;

// 데이터 가져오기

const id = triggerBtn.getAttribute("data-bs-id");

const nickname = triggerBtn.getAttribute("data-bs-nickname");

const body = triggerBtn.getAttribute("data-bs-body");

const articleId = triggerBtn.getAttribute("data-bs-article-id");

console.log("id");

// 데이터를 반영

document.querySelector("#edit-comment-nickname").value = nickname;

document.querySelector("#edit-comment-body").value = body;

document.querySelector("#edit-comment-id").value = id;

document.querySelector("#edit-comment-article-id").value = articleId;

});

}

{

// 수정 완료 버튼 가져오기

const commentUpdateBtn = document.querySelector("#comment-update-btn");

// 클릭 이벤트 감지 및 처리

commentUpdateBtn.addEventListener("click", function() {

// 수정 댓글 객체 생성

const comment = {

id: document.querySelector("#edit-comment-id").value,

nickname: document.querySelector("#edit-comment-nickname").value,

body: document.querySelector("#edit-comment-body").value,

article\_id: document.querySelector("#edit-comment-article-id").value

};

// 수정 REST API 호출 - fetch( ) 사용

const url = "/api/comments/" + comment.id;

fetch(url, {

method: "PATCH", // PATCH 요청

body: JSON.stringify(comment), // 수정된 댓글 객체를 JSON으로 전달

headers: { // body에 담겨진 데이터의 타입 알려주기

"Content-Type": "application/json"

}

}).then(response => {

// http 응답 코드에 따른 메시지 출력

const msg = (response.ok) ? "댓글이 수정 되었습니다." : "댓글 수정 실패!!";

alert(msg);

// 현재 페이지를 새로고침

window.location.reload();

});

});

}

</script>

<!-- 댓글 삭제 -->

<script>

{

// 삭제 버튼 선택

const commentDeleteBtns = document.querySelectorAll(".comment-delete-btn");

// 삭제 버튼 이벤트 처리

commentDeleteBtns.forEach(btn => {

// 선택한 삭제 버튼 요소들이 하나씩 btn으로 들어오고 각 버튼의 이벤트 처리를 등록

btn.addEventListener("click", (e) => {

// 이벤트 발생 요소를 선택

// 클릭 이벤트가 발생된 버튼을 가져옴. e.srcElement == e.target

const commentDeleteBtn = e.srcElement;

// 삭제 댓글 id 가져오기

// 속성이 data-comment-id인 속성을 거져와서 commentId에 넣기

const commentId = commentDeleteBtn.getAttribute("data-comment-id");

console.log(`삭제 버튼 클릭: ${commentId}번 댓글`);

// 삭제 API 호출 및 처리

const url = `/api/comments/${commentId}`;

fetch(url, {

method:"DELETE"

}).then(response => {

// 댓글 삭제 실패 처리

if(!response.ok) {

alert("댓글 삭제 실패!");

return;

}

// 삭제 성공 시, 댓글을 화면에서 지움

const target = document.querySelector(`#comments-${commentId}`);

window.location.reload();

target.remove();

});

});

});

}

</script>