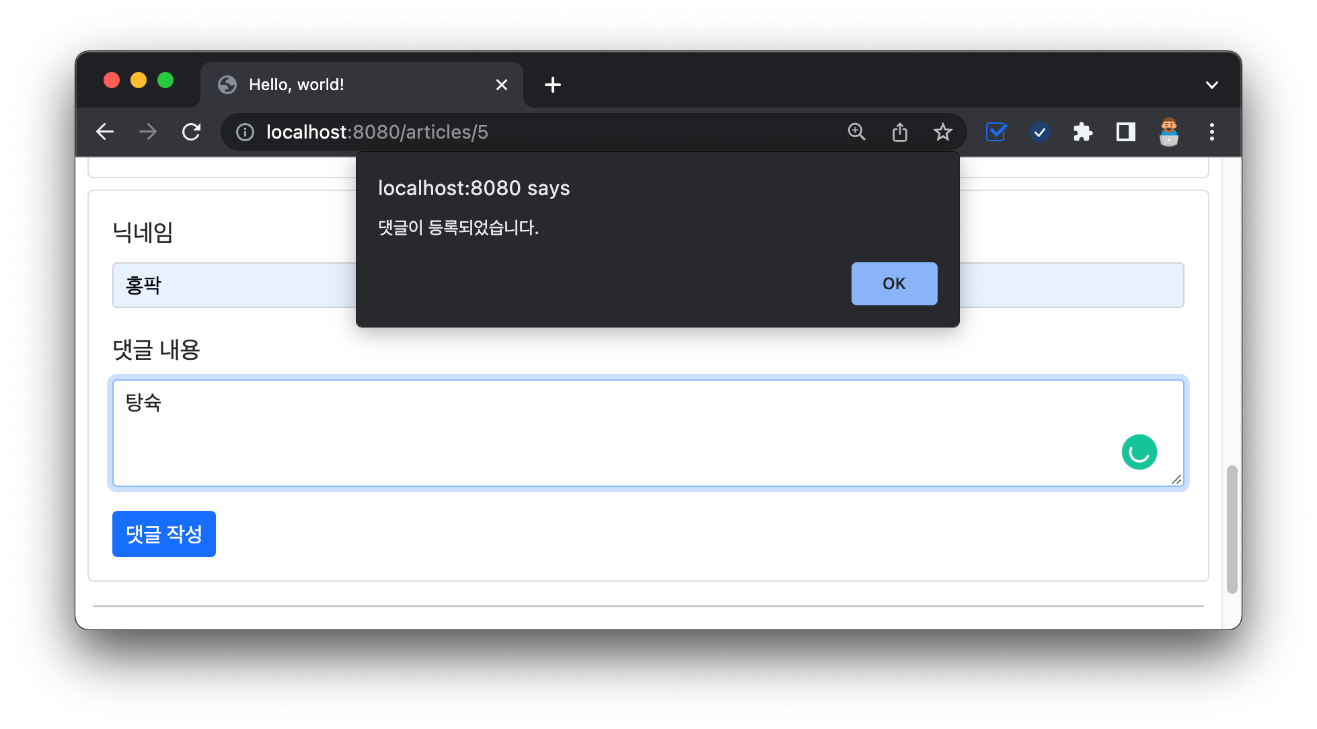
25 댓글 등록 with JS

**댓글 등록 with 자바스크립트(JS)**

**미션**

댓글 등록 페이지를 만들고, REST API를 호출하여 새 댓글을 추가하시오.  


**01:56 댓글 생성 뷰 페이지 - 부트스트랩 Card, Form**

../comments/\_new.mustache

1. <div class="card m-2" id="comments-new">
2. <div class="card-body">
3. <!-- 댓글 작성 폼-->
4. <form>
5. <!-- 닉네임 입력 -->
6. <div class="mb-3">
7. <label class="form-label">닉네임</label>
8. <input type="text" class="form-control form-control-sm" id="new-comment-nickname">
9. </div>
10. <!-- 댓글 본문 입력 -->
11. <div class="mb-3">
12. <label class="form-label">댓글 내용</label>
13. <textarea type="text" class="form-control form-control-sm" rows="3" id="new-comment-body"></textarea>
14. </div>
15. <!-- 히든 인풋 -->
16. {{#article}}
17. <input type="hidden" id="new-comment-article-id" value="{{id}}">
18. {{/article}}
19. <!-- 전송 버튼 -->
20. <button type="button" class="btn btn-outline-primary btn-sm" id="comment-create-btn">댓글 작성</button>
21. </form>
22. </div>
23. </div>

**05:34 버튼 클릭 이벤트 감지**

../comments/\_new.mustache

1. <div class="card m-2" id="comments-new">
2. ...
3. </div>
4. <Script>
5. {
6. // 댓글 생성 버튼 변수화
7. const commentCreateBtn = document.querySelector("#comment-create-btn");
8. // 댓글 클릭 이벤트 감지!
9. commentCreateBtn.addEventListener("click", function() {
10. console.log("버튼이 클릭 되었습니다.");
11. });
12. }
13. </script>

**11:03 새 댓글 JS 객체 생성 - JS 객체 리터럴**

../comments/\_new.mustache

1. <div class="card m-2" id="comments-new">
2. ...
3. </div>
4. <Script>
5. {
6. // 댓글 생성 버튼 변수화
7. const commentCreateBtn = document.querySelector("#comment-create-btn");
8. // 댓글 클릭 이벤트 감지!
9. commentCreateBtn.addEventListener("click", function() {
10. // 새 댓글 객체 생성
11. const comment = {
12. nickname: document.querySelector("#new-comment-nickname").value,
13. body: document.querySelector("#new-comment-body").value,
14. articleId: document.querySelector("#new-comment-article-id").value
15. };
16. // 댓글 객체 출력
17. console.log(comment);
18. });
19. }
20. </script>

**15:34 자바스크립트 REST API 호출 - fetch()**

../comments/\_new.mustache

1. <div class="card m-2" id="comments-new">
2. ...
3. </div>
4. <Script>
5. {
6. // 댓글 생성 버튼 변수화
7. const commentCreateBtn = document.querySelector("#comment-create-btn");
8. // 댓글 클릭 이벤트 감지!
9. commentCreateBtn.addEventListener("click", function() {
10. // 새 댓글 객체 생성
11. const comment = {
12. nickname: document.querySelector("#new-comment-nickname").value,
13. body: document.querySelector("#new-comment-body").value,
14. article\_id: document.querySelector("#new-comment-article-id").value
15. };
16. // 댓글 객체 출력
17. console.log(comment);
18. // fetch() - 비동기 통신을 위한 API
19. const url = "/api/articles/" + comment.article\_id + "/comments";
20. fetch(url, {
21. method: "post",
22. body: JSON.stringify(comment),
23. headers: {
24. "Content-Type": "application/json"
25. }
26. });
27. });
28. }
29. </script>

**22:40 fetch() 응답 처리 - response.ok, alert(), window.location.reload()**

fetch 응답 결과로 화면 바꾸기 - comments/\_new.mustache

1. <div class="card m-2" id="comments-new">
2. ...
3. </div>
4. <Script>
5. {
6. // 댓글 생성 버튼 변수화
7. const commentCreateBtn = document.querySelector("#comment-create-btn");
8. // 댓글 클릭 이벤트 감지!
9. commentCreateBtn.addEventListener("click", function() {
10. // 새 댓글 객체 생성
11. const comment = {
12. nickname: document.querySelector("#new-comment-nickname").value,
13. body: document.querySelector("#new-comment-body").value,
14. article\_id: document.querySelector("#new-comment-article-id").value
15. };
16. // 댓글 객체 출력
17. console.log(comment);
18. // fetch() - 비동기 통신을 위한 API
19. const url = "/api/articles/" + comment.article\_id + "/comments";
20. fetch(url, {
21. method: "post",
22. body: JSON.stringify(comment),
23. headers: {
24. "Content-Type": "application/json"
25. }
26. }).then(response => {
27. // http 응답 코드에 따른 메시지 출력
28. const msg = (response.ok) ? "댓글이 등록되었습니다." : "댓글 등록 실패..!";
29. alert(msg);
30. // 현재 페이지 새로고침
31. window.location.reload();
32. });
33. });
34. }
35. </script>

**🔥 구글링 훈련하기**

* 부트스트랩5 Card
* 부트스트랩5 Form
* HTML form input hidden
* JavaScript addEventListener 클릭
* JavaScript querySelector 사용법
* JavaScript 객체 리터럴
* JavaScript console log
* JavaScript fetch
* JavaScript alert
* JavaScript location reload