19 장 완성 코드

comments/_list.mustache

```
<div id="comments-list">
   {{#commentDtos}}
       ⟨div class="card m-2" id="comments-{{id}}"⟩
           <div class="card-header">
               {{nickname}}
               <!-- Button trigger modal -->
               class="btn btn-sm btn-outline-primary"
                       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 -->
           class="btn btn-sm btn-outline-danger comment-delete-btn"
                   data-comment-id="{{id}}">삭제</button>
           </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"</pre>
                       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"</pre>
                              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" rows="3"</pre>
                               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 -->
                   Soutton type="button" class="btn btn-primary"
                           id="comment-update-btn">수정 완료〈/button〉
               </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(event) {
       // 1. 트리거 버튼 선택
       const triggerBtn = event.relatedTarget;
       // 2. 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");
       // 3. 수정 폼에 데이터 반영
       document.guerySelector("#edit-comment-nickname").value = nickname;
       document.guerySelector("#edit-comment-body").value = body;
       document.guerySelector("#edit-comment-id").value = id;
       document.guerySelector("#edit-comment-article-id").value = articleId;
   });
}
   // 수정 완료 버튼 선택
   const commentUpdateBtn = document.querySelector("#comment-update-btn");
```

```
// 클릭 이벤트 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.guerySelector("#edit-comment-body").value,
          article_id: document.guerySelector("#edit-comment-article-
id").value
       };
       console.log(comment);
       // 수정 REST API 호출
       const url = "/api/comments/" + comment.id;
       fetch(url, {
          method: "PATCH",
                            // PATCH 요청
          headers: {
                             // 전송 데이터 타입(JSON) 정보
              "Content-Type": "application/json"
          body: JSON.stringify(comment) // JSON 문자열로 변환하여 전송
       }).then(response => {
       // HTTP 응답 코드에 따른 메시지 출력
       const msg = (response.ok) ? "댓글이 수정됐습니다." : "댓글 수정 실
패..!";
       alert(msg);
       // 현재 페이지 새로고침
       window.location.reload();
       });
   });
</script>
<!-- 댓글 삭제 -->
<script>
   // 삭제 버튼 선택
   const commentDeleteBtns = document.querySelectorAll(".comment-delete-
btn");
   // 삭제 버튼 이벤트 처리
   // commentDeleteBtn.addEventListener("click", function() {
   // console.log("삭제 버튼이 클릭됐습니다..!");
   // });
   commentDeleteBtns.forEach(btn => {
       btn.addEventListener("click", (event) => {
          // 이벤트 발생 요소 선택
          const commentDeleteBtn = event.target;
          // 삭제 댓글 id 가져오기
          const commentId = commentDeleteBtn.getAttribute("data-comment-id");
          console.log(`삭제 버튼 클릭: ${commentId}번 댓글`);
          // 삭제 REST API 호출
          const url = `/api/comments/${commentId}`;
```

```
fetch(url, {
              method: "DELETE"
          }).then(response => {
              // 댓글 삭제 실패 처리
              if (!response.ok) {
                 alert("댓글 삭제 실패..!");
                 return;
              }
              // 삭제 성공 시 댓글을 화면에서 지우고 메시지 창 띄우기
              const target = document.querySelector(`#comments-${commentId}`);
              target.remove();
              const msg = `${commentId}번 댓글을 삭제했습니다`;
              alert(msg);
              // 현재 페이지 새로고침
              window.location.reload();
          });
      });
  });
</script>
```

셀프체크 정답

resources/static/selfcheck19.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   〈title〉셀프체크 19〈/title〉
</head>
<body>
 <h1>Hello, World!</h1>
 <u1>
   HTML
   ⟨li class="prerequisite"⟩CSS⟨/li⟩
   ⟨li class="prerequisite"⟩JavaScript⟨/li⟩
   ⟨li class="prerequisite"⟩Java⟨/li⟩
   SQL
   Spring Boot
 <script>
 const nodeList = document.querySelectorAll(".prerequisite");
 nodeList.forEach((item) => {
   item.style.textDecoration = "line-through";
 });
 </script>
</body>
</html>
```