18 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 -->
               <button type="button"</pre>
                      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>
           </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">
               댓글 수정〈/h5〉
               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 -->
                  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.querySelector("#edit-comment-nickname").value = nickname;
       document.guerySelector("#edit-comment-body").value = body;
       document.querySelector("#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.guerySelector("#edit-comment-id").value,
           nickname: document.querySelector("#edit-comment-nickname").value,
           body: document.querySelector("#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>
```

comments/ new.mustache

```
<div class="card m-2" id="comments-new">
    <div class="card-body">
        <!-- 댓글 작성 폼 -->
        <form>
            <!-- 닉네임 입력 -->
            <div class="mb-3">
                <label class="form-label">닉네임</label>
                <input type="text" class="form-control"</pre>
                      id="new-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="new-comment-body"></textarea>
            </div>
            <!-- 히든 인풋 -->
            {{#article}}
                <input type="hidden" id="new-comment-article-id"</pre>
                      value="{{id}}}">
```

```
{{/article}}
          <!-- 전송 버튼 -->
          id="comment-create-btn">댓글 작성</button>
      </form>
   </div>
</div>
<script>
   // 댓글 생성 버튼 변수화
   const commentCreateBtn = document.guerySelector("#comment-create-btn");
   // 댓글 클릭 이벤트 감지
   commentCreateBtn.addEventListener("click", function() {
      // 새 댓글 객체 생성
      const comment = {
          // 새 댓글의 닉네임
          nickname: document.guerySelector("#new-comment-nickname").value,
          // 새 댓글의 본문
          body: document.querySelector("#new-comment-body").value,
          // 부모 게시글의 id
          articleId:
                           document.guerySelector("#new-comment-article-
id").value
      };
      // 댓글 객체 출력
      console.log(comment);
      // fetch() - 비동기 통신을 위한 API
      const url = "/api/articles/" + comment.articleId + "/comments";
      fetch(url, {
          method: "POST", // POST 요청
                         // 전송 본문의 데이터 타입(JSON) 정보
          headers: {
          "Content-Type": "application/json"
          body: JSON.stringify(comment) // JSON 문자열로 변환해 전송
      }).then(response => {
          // HTTP 응답 코드에 따른 메시지 출력
          const msg = (response.ok) ? "댓글이 등록됐습니다." :
                                  "댓글 등록 실패..!";
          alert(msg);
          // 현재 페이지 새로고침
          window.location.reload();
      });
   });
}
</script>
```

셀프체크 정답

resources/static/selfcheck18.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  〈title〉셀프체크 18〈/title〉
                          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <script defer</pre>
         src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js">
  </script>
</head>
<body>
  <h1>모달로 데이터 전달</h1>
  <!-- 모달 트리거 버튼 -->
  <button id="trigger-btn"</pre>
         data-bs-toggle="modal"
         data-bs-target="#exampleModal"
         data-custom-title="점심 뭐 먹지?"
         data-custom-text="돈가스 어때?">트리거 버튼〈/button〉
  <!-- 모달 -->
  <div class="modal fade" id="exampleModal" tabindex="-1">
    ⟨div class="modal-dialog"⟩
      <div class="modal-content">
        <div class="modal-header">
          \(\h1 class=\)modal-title fs-5\) id=\)modal-title\(\hat{\chi}\)???\(\h1\)
          Solution
                        type="button" class="btn-close"
                                                                     data-bs-
dismiss="modal"></button>
        </div>
        <div class="modal-body">
         \( p id="modal-text"\)...\( / p \)
        </div>
      </div>
   </div>
  </div>
  <script>
   // 모달 가져오기
   const modalDiv = document.querySelector("#exampleModal");
   // 모달 이벤트 처리
   modalDiv.addEventListener("show.bs.modal", function(event){
   // 트리거 버튼
```

```
const triggerBtn = event.relatedTarget;

// 변경 대상 가져오기
const modalTitle = document.querySelector("#modal-title");
const modalText = document.querySelector("#modal-text");

// 전달값 가져오기
const title = triggerBtn.getAttribute("data-custom-title");
const text = triggerBtn.getAttribute("data-custom-text");

// 값 변경
modalTitle.innerText = title;
modalText.innerText = text;
});
</script>
</body>
</html>
```