

스프링 부트 웹 프로젝트

chapter09

AJAX로 댓글 처리

제공된 자료는 훈련생의 수업을 돕기 위한 것으로, 타인과 공유하시면 안됩니다.

Contents

part.1

AJAX 개념

part.2

댓글 관리 기능

part.3

게시물 상세화면에서 댓글 처리

해당 프로젝트는 기존 코드를 활용할 예정입니다.

 project08

이전에 작성한 "project8"을 복사하여 "project9"를 새로 생성하세요.

 project09

프로젝트에서 에러가 발생했으면, setting.gradle 파일을 열어서 프로젝트의 이름을 변경하세요.

그리고 gradle을 리프레쉬하면 에러가 사라집니다.

```
▼ > project09 [spring-study main]
  > > src/main/java
  > > src/main/generated
  > > src/main/resources
  > > src/test/java
```

```
settings.gradle ×
1 rootProject.name = 'project8'
2
```



```
settings.gradle ×
1 rootProject.name = 'project09'
```

프로젝트가 정상적으로 실행되는지 확인하세요.

```

    .
   /\ /__ \ ,      ( )          __\__ \ 
( ( ) \__ \| ' _ | ' _ | ' _ | \_ \| \_ \| 
\\ /   ___) | |_) | | | | | | | ( | | ) ) ) 
'   |___| | ._| | | | | | | \_ \| / / / / 
=====|_|=====|___|_/=/_/_/_/_/

:: Spring Boot ::                (v3.3.0)

2024-06-12T19:22:21.206+09:00 INFO 129672 --- [main] com.example.demo.Project
2024-06-12T19:22:21.208+09:00 INFO 129672 --- [main] com.example.demo.Project
2024-06-12T19:22:21.765+09:00 INFO 129672 --- [main] .s.d.r.c.RepositoryConf
2024-06-12T19:22:21.820+09:00 INFO 129672 --- [main] .s.d.r.c.RepositoryConf
2024-06-12T19:22:22.191+09:00 INFO 129672 --- [main] o.s.b.w.embedded.tomcat.
2024-06-12T19:22:22.213+09:00 INFO 129672 --- [main] o.apache.catalina.core.S

```

AJAX는 웹페이지에서 클라이언트와 서버 간에 비동기식으로 데이터를 주고받는 기술이다.

데이터는 주로 XML이나 JSON 형식으로 데이터를 주고 받는다.

예를 들어 사용자가 "중복체크" 버튼을 클릭하면 서버가 중복 여부를 확인한 후 클라이언트에게 결과를 전송하고, 화면에서 필요한 부분만 업데이트한다.

2.화면에 결과 메시지 표시

1.서버에 아이디 중복 여부를 요청

Sign Up

아이디가 중복되었습니다..

아이디

중복체크

패스워드

이름

☒ 일반사용자
☐ 관리자

등록

회원 가입 화면

```
{  
  ...  
  "msg": "아이디가중복되었습니다"  
}
```

전송 받은 데이터

AJAX는 비동기 통신을 사용하여 페이지 전체를 새로고침 하지 않고 필요한 부분만 업데이트할 수 있어 응답 속도가 빠르다.
그래서 서버의 응답을 기다리는 동안에도 사용자는 화면의 다른 기능을 자유롭게 사용할 수 있다.
반면, 동기식 통신은 서버 응답을 기다리는 동안 화면이 멈추기 때문에 다른 기능을 사용할 수 없다는 단점이 있다.

Sign Up

아이디가 중복되었습니다..

아이디
user1

패스워드
.....

이름
도우너

☒ 일반사용자
☐ 관리자

등록

중복체크

응답 대기중...

동기식 처리

Sign Up

아이디가 중복되었습니다..

아이디
user1

패스워드
.....

이름
도우너

☒ 일반사용자
☐ 관리자

등록

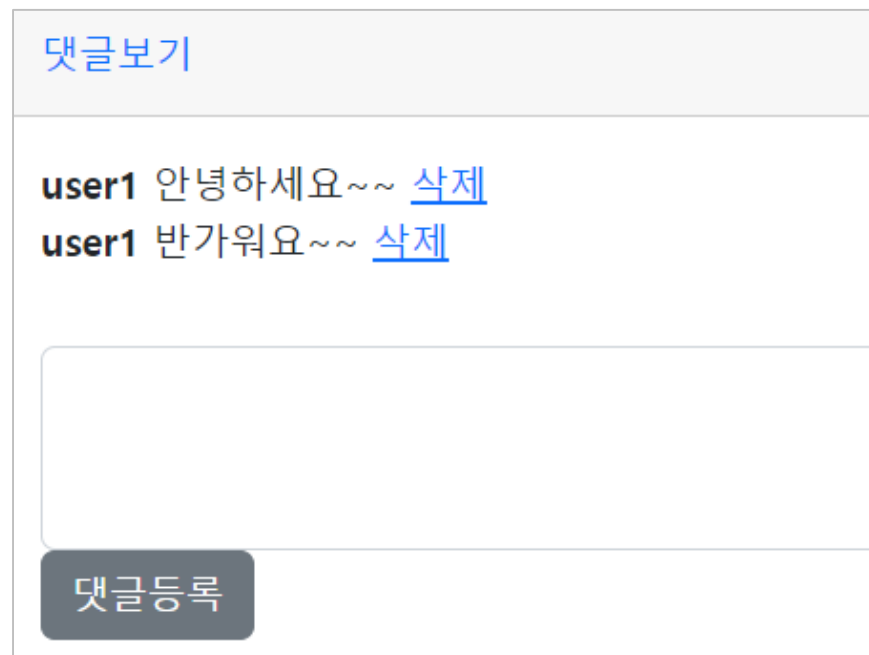
중복체크

응답 대기중...

비동기식 처리

AJAX는 다양한 기능에 활용할 수 있다.

예를 들어 게시판에서 댓글을 달 때 댓글 목록을 실시간으로 업데이트할 수 있고,
채팅 프로그램에서 메시지를 주고 받을 때 대화 내용을 실시간으로 업데이트할 수 있다.

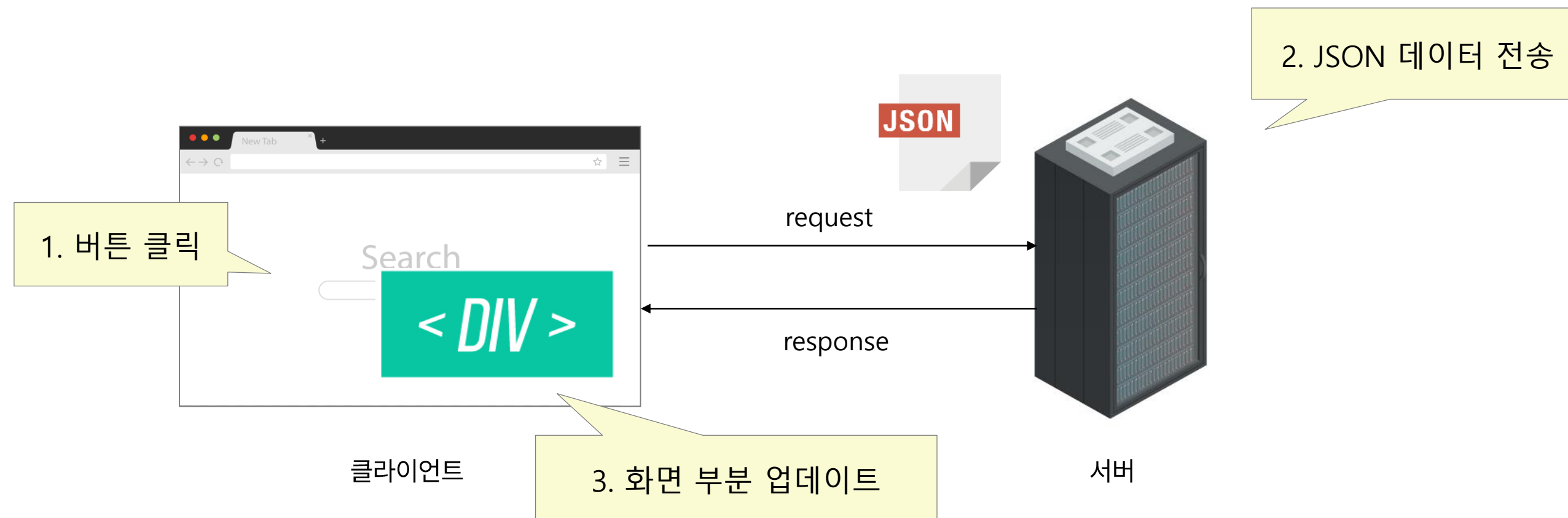


댓글 기능



채팅 기능

1. 사용자가 AJAX로 서버에 데이터를 요청한다.
2. 서버는 요청을 처리하고, 응답 데이터를 클라이언트에게 전송한다.
3. 클라이언트는 전달받은 데이터를 사용하여 화면의 특정 부분만 업데이트한다.



이번에는 AJAX를 소개하기 위해 호출 부분만 jQuery를 사용하고 나머지 코드는 바닐라 자바스크립트를 사용하겠습니다. 이전에 바닐라 자바스크립트의 fetch를 배웠지만, 이번에는 스프링 교재에서 사용하고 있는 jQuery 라이브러리의 AJAX를 사용하겠습니다. fetch와 AJAX 모두 서버와 통신할 때 사용하는 함수입니다.

바닐라 자바스크립트

순수한 자바스크립트를 의미한다.

외부 라이브러리 없이 자바스크립트 코드를 작성하는 방식이다.

jQuery

자바스크립트를 쉽게 사용할 수 있게 도와주는 라이브러리이다.

DOM 조작, 이벤트 처리, AJAX 요청 등을 처리할 수 있도록 다양한 기능을 제공한다.

```
<script src="https://code.jquery.com/jquery-3.4.1.js"></script>  
head>
```

댓글 관리 기능

댓글 관리 기능

사용자는 게시물 상세 페이지에서 댓글 목록을 확인하고, 새로운 댓글을 등록하거나 기존 댓글을 삭제할 수 있다.

Board Read Page

번호
1

제목
제목~

내용
내용~

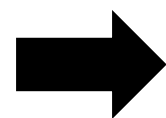
작성자
us

등록
20

수정
20

댓글보기

댓글등록



Board Read Page

번호
1

제목
제목~

내용
내용~

작성자
us

등록
20

수정
20

댓글보기

user1 안녕하세요~ [삭제](#)

user1 반갑습니다~ [삭제](#)

댓글등록

댓글 엔티티를 생성하고 BaseEntity를 상속받는다.

댓글 리파지토리는 JpaRepository를 상속받아 댓글 데이터를 처리할 수 있도록 설정한다.

Comment

```
public class Comment extends BaseEntity {
```

CommentRepository

```
public interface CommentRepository extends JpaRepository<Comment, Integer> {
```

Q. 단위테스트와 리파지토리를 이용하여 댓글을 등록, 조회, 수정, 삭제 한다.

CommentRepositoryTest

```
@Test
public void 댓글등록() { }

@Test
public void 댓글목록조회() { }

@Test
public void 댓글단건조회() { }

@Test
public void 댓글수정() { }

@Test
public void 댓글삭제() { }
```

리파지토리에 댓글 처리

- 특정 게시물에 달린 모든 댓글을 조회하는 메소드를 추가한다.
- 특정 게시물에 달린 모든 댓글을 삭제하는 메소드를 추가한다.
- 삭제 메소드를 호출하면 트랜잭션 처리가 거부되어 에러가 발생한다.

이 문제는 클래스나 메소드에 @Transactional 어노테이션을 추가하면 정상적으로 처리된다.

CommentRepository

```
List<Comment> findByBoard(Board board);
```

```
@Transactional
```

```
void deleteByBoard(Board board);
```

에러 메시지

```
: No EntityManager with actual transaction available for current thread - cannot reliably process 'remove' call
```

Q. 단위테스트와 리파지토리를 이용하여 게시물별로 댓글을 조회하고 삭제한다.

CommentRepositoryTest

```
@Test
public void 게시물별_댓글조회() {

}

@Test
public void 게시물별_댓글삭제() {

}
```

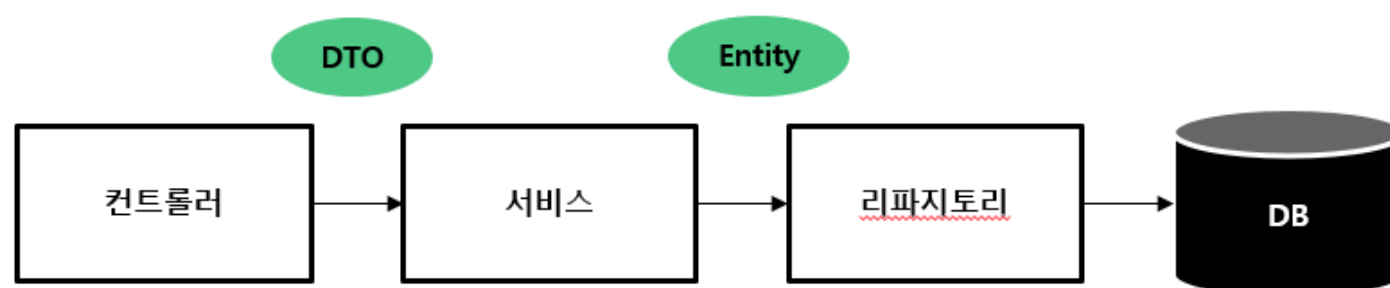
CommentDTO

컨트롤러와 뷰 사이에서 댓글 데이터를 전달하는 객체를 생성한다.

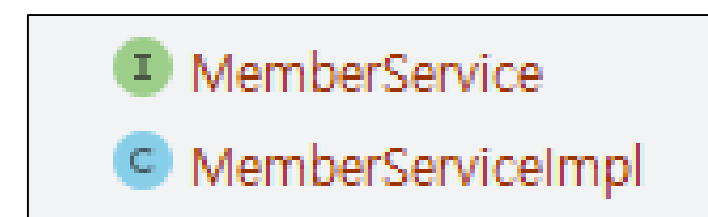
CommentService와 CommentServiceImpl

컨트롤러와 리파지토리 사이에서 사용자 요청을 전달하는 서비스를 생성한다.

서비스는 인터페이스와 구현클래스로 구성되어 있다.



서비스 구조



서비스에서 댓글 처리

- 게시물에 달린 댓글 목록을 조회하는 메소드를 추가한다.
- 새로운 댓글을 등록하는 메소드를 추가한다.
- 특정 게시물에 달린 댓글을 삭제하는 메소드를 추가한다.
- DTO와 Entity 간의 변환 메소드를 추가한다.

인터페이스

```
List<CommentDTO> getListByBoardNo(int boardNo);  
  
int register(CommentDTO dto);  
  
void remove(int no);  
  
...
```

구현 클래스

```
@Service  
public class CommentServiceImpl implements CommentService {  
  
    @Override  
    public int register(CommentDTO dto) {  
  
        ...  
    }  
}
```


Q. 단위테스트와 서비스를 이용하여 댓글을 등록, 조회, 삭제 한다.

CommentServiceTest

```
@Test
public void 댓글등록() { }

@Test
public void 게시물별_댓글목록조회() { }

@Test
public void 댓글삭제() { }
```

- 댓글 목록 조회
- 댓글 등록
- 댓글 수정 (생략)
- 댓글 삭제

URL 설계

기능	URL	GET/POST	결과 데이터	설명
목록	/comment/list	GET	Comment DTO 리스트	게시물별 댓글 목록 조회
등록	/comment/register	POST	등록 성공 여부	댓글 등록 처리
삭제	/comment/remove	DELETE	삭제 성공 여부	댓글 삭제 처리

컨트롤러는 AJAX 호출을 처리하기 위해 API 형식으로 작성한다.

API는 화면없이 데이터만 처리하는 프로그램이다.

@RestController는 @Controller와 @ResponseBody를 결합한 어노테이션이다.

이 어노테이션을 사용하면 응답데이터가 JSON 형식으로 반환된다.

CommentController

```
@RestController
@RequestMapping("/comment")
public class CommentController {
```

- 게시물 번호를 기준으로 댓글 목록을 조회하는 메소드를 추가한다.
게시물 번호를 파라미터로 받아 처리한 후, 댓글 목록을 반환한다.
- 새로운 댓글을 등록하는 메소드를 추가한다.
댓글 정보를 파라미터로 받아 처리한 후, 등록 성공 여부를 반환한다.
- 댓글을 삭제하는 메소드를 추가한다.
댓글 번호를 파라미터로 받아 처리한 후, 삭제 성공 여부를 반환한다.

CommentController

```
@GetMapping("/list")
public List<CommentDTO> list(int boardNo) {
}
@PostMapping("/register")
public Boolean register(CommentDTO dto) {
}
@GetMapping("/remove")
public Boolean remove(int commentNo) {
}
```

댓글 보기란 만들기

- 게시물 상세 화면에 댓글보기란을 추가하여, 댓글 목록과 입력칸을 표시한다.
- 댓글 입력 후 버튼을 클릭하면 등록 요청을 보낸다.

read.html

```
<!-- 댓글 -->
<div class="card shadow mb-4" style="margin-top:10px">
  <div class="card-header py-3">
    <h6 class="m-0 font-weight-bold text-primary">댓글보기</h6>
  </div>
  <div class="card-body" id="commentdiv">
  </div>
  <div class="card-body">
    <textarea class="form-control" id="commentContent" rows="3">
    </textarea>
    <button type="button" class="btn btn-secondary" onclick="">댓글등록</button>
  </div>
</div>
<!-- 댓글 끝 -->
```

PPT에서 코드를 복사하여
수정일 필드 아래에 추가하세요

수정일
2024/01/30 19:59:50

댓글보기

댓글등록

수정 등록

조회 화면에서 댓글 처리 댓글 리스트 보여주기

- 댓글은 AJAX로 서버에 요청하고, JSON형식의 commentDto 리스트를 받아 처리한다.
- 페이지가 만들어질 때, 자동으로 함수가 호출되도록 처리한다.
- 서버에서 받은 댓글 목록으로 HTML 태그를 생성하여 화면에 표시한다.

read.html

```
function showComment() {  
    $.ajax({  
        type: " 메소드종류",  
        url: " 댓글목록요청주소",  
        success: function(response) {  
            서버에서 응답을 받은 후 댓글 목록을 처리하는 코드  
        }  
    });  
}  
// 화면이 생성될 때 함수 호출  
$(document).ready(function() {  
    showComment();  
})
```

조회 화면에서 댓글 처리 댓글 리스트 보여주기

테스트

1. 먼저 단위테스트를 통해 댓글을 등록한다.
2. 게시물 상세 화면으로 이동한다.
3. 상세화면에서 댓글 목록이 나타나는지 확인한다.

수정일

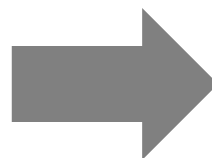
2024/01/30 19:59:50

댓글보기

댓글등록

수정

목록



댓글보기

user1 댓글입니다 [삭제](#)

조회 화면에서 댓글 처리 댓글 등록하기

read.html

- 댓글 등록 버튼을 클릭하면 서버에 등록 요청을 보내는 이벤트를 처리한다.
- 게시물 번호와 댓글 내용을 JSON 데이터로 만들어서 서버로 전송한다.
- 등록이 성공하면, 댓글조회 메소드를 호출하여 댓글 목록을 갱신한다.

read.html

```
function registerComment () {  
    $.ajax({  
        type: "메소드종류",  
        data: "JSON데이터",  
        url: "등록처리URL",  
        success: function(response) {  
            댓글 등록 후 화면을 갱신하는 코드  
        }  
    });  
}
```

read.html

```
<button onclick="registerComment()">댓글등록</button>
```


조회 화면에서 댓글 처리 댓글 등록하기

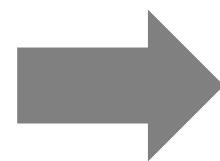
테스트

1. 게시물 상세 화면으로 이동한다.
2. 새로운 댓글을 등록한다.
3. 댓글 목록이 갱신되는지 확인한다.

댓글보기

안녕하세요

댓글등록



댓글보기

aa 안녕하세요 [삭제](#)

- 댓글 삭제 버튼을 클릭하면 서버에 삭제 요청을 보내는 이벤트를 처리한다.
- 댓글 번호를 서버로 전송한다.
- 삭제가 성공하면 화면에서 댓글 목록을 갱신한다.

read.html

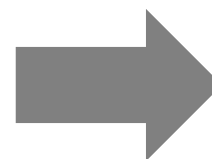
```
<script th:inline="javascript">
    function removeComment (commentNo) {
        $.ajax({
            type: "",
            url: "",
            success: function (response) {
            }
        });
    }
</script>
```

조회 화면에서 댓글 처리 댓글 삭제하기

테스트

1. 게시물 상세 화면으로 이동한다.
2. [삭제] 버튼을 클릭하여 댓글을 삭제한다.
3. 댓글이 사라지는지 확인한다.

댓글보기
aa 안녕하세요 삭제



댓글보기
댓글등록