



# 게시물 출력 및 게시물 삭제

Web 개발 기초





1. 게시물 출력

2. 게시물 삭제



# 1. 게시물 출력

## DB에서 데이터 받아오기

먼저 route.js 를 수정해준다

Route.js에서 '/' 부분의 내용을

우측과 같이 수정해준다.

```
app.get('/', function(req, res){
  var db = mysql.createConnection(db_config);
  if(typeof req.user == 'undefined'){
    console.log('Unauthorized access-main');
    res.redirect('/login');
  } else {
    var sql = 'SELECT * FROM `post` WHERE 1';
    db.query(sql, function(err, rows){
      if(err){
        console.error('mysql connection error');
        console.error(err);
        throw err;
      }
      var posts = rows;
      res.render('index', {
        title: "insta_project",
        username: req.user.user_id,
        posts: posts
      });
    });
  }
  db.end();
});
```



# 1. 게시물 출력

## 템플릿에 출력

그리고 index.ejs 에서 main-body 의 내부 내용을 수정해준다.

이 때 for문을 잘 닫아주도록 한다.



# 1. 게시물 출력

## 템플릿에 출력

```
<div class="main-body">
  <% for (var i=0;i<posts.length;i++) { %>
  <div class="post">
    <div class="post-header">
      
      <div class="author-id">
        <%=posts[i].user_id%>
      </div>
      <div class="post-reg-time">
        <%=posts[i].reg_date%>
      </div>
    </div>
    <div class="post-body" style="text-align:center;">
      
      <span><%=posts[i].content%></span>
    </div>
    <div class="post-footer">
      <div class="num-of-comment">
        댓글 20개
      </div>
      <div class="comment-list">
        <div class="each-comment">
          <div class="userId">
            Mark
          </div>
          <div class="comment">
            I like this
          </div>
        </div>
        <div class="each-comment">
          <div class="userId">
            Kong
```

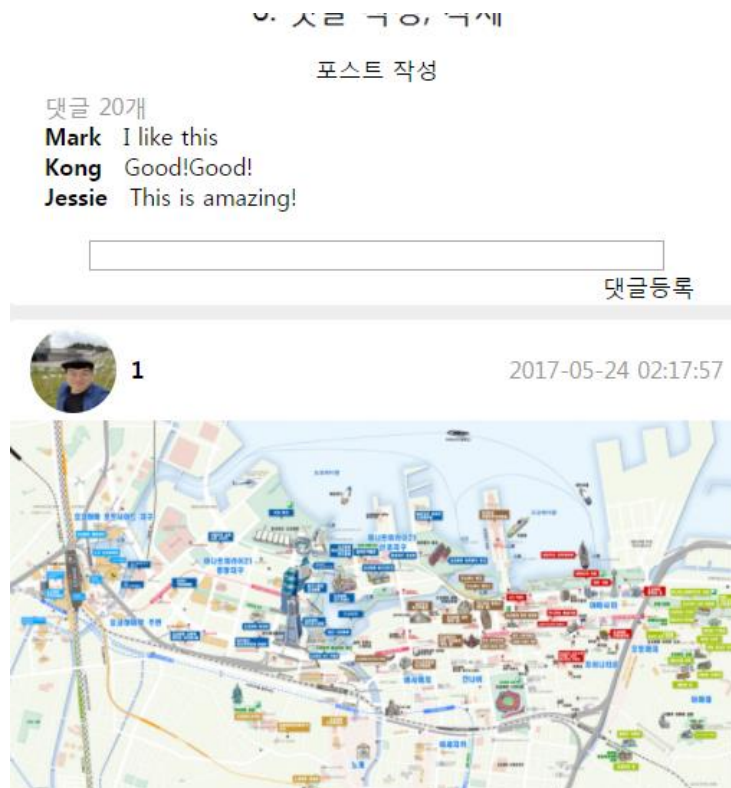
```
</div class="comment-button">
      댓글등록
    </div>
  </div>
</div>
<% } %>
</div>
<div class="footer">
  <div class="footer-container">
```

# 1. 게시물 출력

## 템플릿에 출력

이와 같이 작성하고 브라우저를 확인해보면

내가 등록했던 게시물들이 쪽 나열되는 것을 확인할 수 있다.





# 1. 게시물 출력

## 템플릿에 출력

하지만 인스타는 최신이 가장 위로 오기 때문에 우리도 그렇게 해준다.

Route.js 파일에서 '/' 부분의 sql 을 아래와 같이 변경

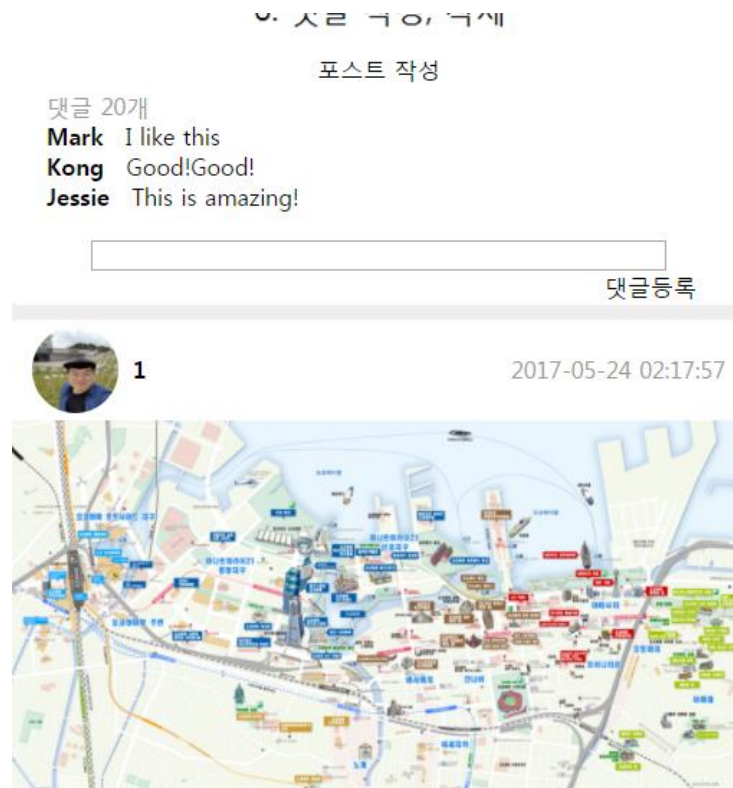
```
res.redirect('/login');  
} else {  
  var sql = 'SELECT * FROM `post` ORDER BY `idx` DESC';  
  db.query(sql, function(err, rows){  
    if(err){
```

# 1. 게시물 출력

## 템플릿에 출력

이와 같이 작성하고 브라우저를 확인해보면

내가 등록했던 게시물들이 시간 순서로 쭉 나열되는 것을 확인할 수 있다.





## 2. 게시물 삭제

### 삭제 버튼 생성 및 삭제 기능 구현

먼저 삭제를 위해 삭제버튼을 만들어준다.

Post 의 body 부분에 삭제버튼을 만든다.

```
<span><%=posts[i].content%></span>
<div>
  <input type="button" class="post-delete-btn" value="삭제" data-idx="<%=posts[i].idx%>" />
</div>
</div>
<div class="post-footer">
```

## 2. 게시물 삭제

### 삭제 버튼 생성 및 삭제 기능 구현

브라우저에서 확인해보면 버튼이 생성된 것을 확인할 수 있다.

스타일은 이상하지만 기능구현에 초점을 두고 일단 넘어간다.





## 2. 게시물 삭제

### 삭제 버튼 생성 및 삭제 기능 구현

삭제 버튼을 눌렀을 때의 기능을 구현하기 위해 index.ejs의 맨 아래 Body 윗 부분에 아래와 같이 작성해준다.

```
<script>
$( '.post-delete-btn' ).click(function(){
  if(confirm("삭제하시겠습니까")){
    var idx = $(this).data('idx');
    location.href="/delete?idx="+idx;
  } else {
    return;
  }
});
</script>
</body>
</html>
```



## 2. 게시물 삭제

### DB에서 삭제 및 파일 삭제

이제 버튼을 만들었으니 서버측을 작성해 DB내용을 삭제하고,

업로드된 파일을 삭제하도록 한다.

먼저 route.js 에서 delete 부분을 작성한다.

## 2. 게시물 삭제

### DB에서 삭제 및 파일 삭제

```
app.get('/delete', function(req, res){
  var db = mysql.createConnection(db_config);
  var idx = req.query.idx;
  var sql = 'DELETE FROM `post` WHERE `idx` = ?';
  db.query(sql, [idx], function(err, rows){
    if(err){
      console.error('mysql connection error');
      console.error(err);
      throw err;
    }
  });
  var filename = req.query.filename;
  var path_name = './static/img/uploads/' + filename;
  fs.unlink(path_name,
    function(err){
      if(err) throw err;
      console.log('파일을 정상적으로 삭제하였습니다.');
```

## 2. 게시물 삭제

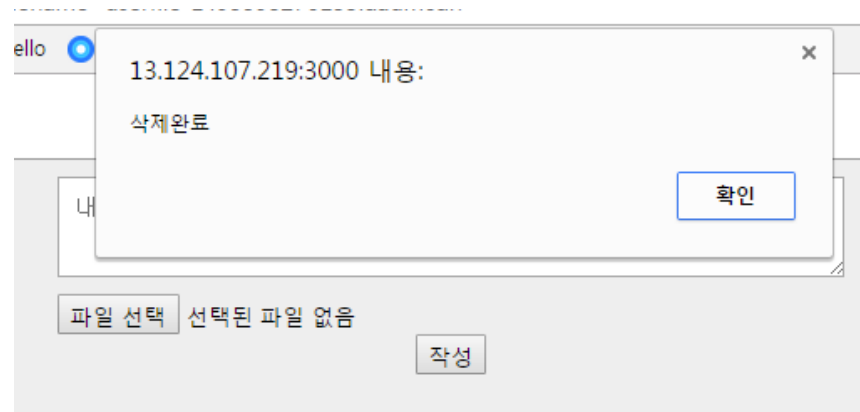
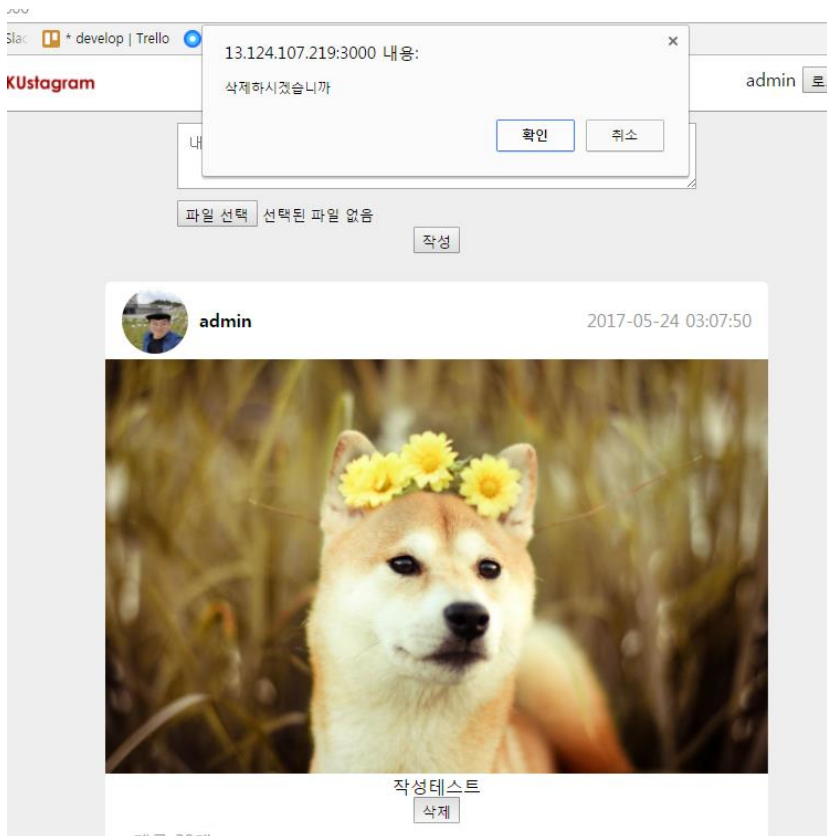
그리고 게시물을 삭제하는 버튼은 작성한 사람만 보일 수 있게

삭제 버튼 아래 위에 if 조건문을 만들어 준다.

```
<% if(posts[i].user_id==username) { %>
<div>
  <input type="button" class="post-delete-btn" value="삭제"
  data-idx="<%=posts[i].idx%" data-filename="<%=posts[i].img_src%" />
</div>
<% } %>
```

## 2. 게시물 삭제

브라우저에서 확인해 보면 삭제가 잘 되는 것을 확인할 수 있다.





**수업 끝!**