



# [프로젝트] 1. 새가 날아든다

💡 140자 이내 익명 인스턴트 메모 서비스입니다. 이번 시간을 통해 다음 내용을 배울 수 있습니다.

- 1) 익명으로 메모 작성하기
- 2) 최신 순으로 메모 정렬하고 타임라인에 보여주기
- 3) 24시간이 지난 데이터는 보여주지 않기
- 4) 메모 수정하기

🎬 [\[새가날아든다\] 강의영상 모음\(클릭\)](#)

## 1. [새가날아든다] - 프로젝트 세팅

🔥 pycharm의 new project를 클릭해서,  
sparta → project → birdcomming 폴더를 만들고 시작하기!

- ▼ 1) 완성작보기
  - 링크를 통해 살펴보기
- ▼ 2) 패키지 설치하기

설치할 패키지: flask pymongo

JavaScript ▾

## ▼ 3) 폴더 세팅하고 &amp; 시작 코드 붙여넣기

✓ static, templates 폴더 + app.py 이제 너무 익숙하죠~? 😊

## ▼ index.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>
코딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap
integrity="sha384-
Vko08x4CGs03+Hhvxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"> <script src="https://code.jquery.com/jquery-
3.4.1.slim.min.js" integrity="sha384-
J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.m
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
</script> <script> function openclose() { let status = $('#post-box').css('displ
(status === 'block') { $('#post-box').hide(); } else { $('#post-box').
} </script> <style> .area-edit { display: none; } .wrap { width: 700px;
10px auto; } #contents { width: 700px; } .area-write { display: flex;
direction: column; align-items: flex-end; justify-content: center; } .
card { margin: 20px 0 20px 0; } #post-box { display: none; } </style>
<body> <div class="wrap"> <div class="header"> <h1>새가 날아든다</h1> <
<button onclick="openclose()" type="button" class="btn btn-outline-info
sm">공유하고 싶은 소식</button>을 입력해주세요. 24시간이 지난 뒤에는 사라
</p> </div> <div id="post-box"> <div class="area-write"> <textarea
placeholder="공유하고 싶은 소식을 입력해주세요" name="contents" id="cont
cols="30" rows="10"></textarea> <button class="btn btn-danger"
onclick="writePost()">작성하기</button> </div> </div> <div id="cards-bo
class="area-read"> <div class="card"> <header class="card-header"> <p
class="card-header-title"> Component </p> <a href="#" class="card-head
aria-label="more options"> <span class="icon"> <i class="fas fa-angle-r
aria-hidden="true"></i> </span> </a> </header> <div class="card-conten
class="content"> Lorem ipsum dolor sit amet, consectetur adipiscing el
Phasellus nec iaculis mauris. <a href="#">@bulmaio</a>. <a href="#">#c
href="#">#responsive</a> <br> <time datetime="2016-1-1">11:09 PM - 1 J
2016</time> </div> </div> <footer class="card-footer"> <a href="#" cla
```

```

        footer-item">Save</a> <a href="#" class="card-footer-item">Edit</a> <a
        class="card-footer-item">Delete</a> </footer> </div> </div> </div> </b>
    </html>

```

### ▼ app.py

```

from flask import Flask, render_template, request, jsonify from
pymongo import MongoClient from datetime import datetime, timedelta
app = Flask(__name__) # URL 별로 함수명이 같거나, # route('/') 등의
주소가 같으면 안됩니다. client = MongoClient('localhost', 27017) db =
client.bird_comming @app.route('/') def home(): return
render_template('index.html') if __name__ == '__main__':
app.run('0.0.0.0', port=5000, debug=True)

```

Python ▼

## 2. [새가날아든다] - 익명으로 메모 작성하기

## ▼ 4) 익명으로 메모 작성 기능 만들기

- ✓ 익명으로 메모를 작성하기 위해서는 다음 요소가 필요합니다.

**[클라이언트]**

- 1) 사용자가 작성한 메모 내용을 가져와서 올바른 입력인지 확인한다.
- 2) 랜덤으로 유저이름을 생성한다.
- 3) 유저이름과 메모 내용을 서버(/message)에 전달한다.

**[서버]**

- 1) 유저이름과 메모를 받아 DB에 저장하기

## ▼ (1) 클라이언트 코딩

- ▼ 1: 사용자가 작성한 메모 내용을 가져와서 올바른 입력인지 확인한다.

```
function isValidContents(contents) { // 1. 사용자가 입력을 하였는
  지 확인한다. if (contents == '') { alert('내용을 입력해주세요');
  return false; } // 2. 사용자의 입력 내용이 141자 이상인지 확인한
  다. if (contents.trim().length > 140) { alert('공백 포함 140자 이
  하로 입력해주세요'); return false; } // 3. 올바른 입력일 경우 true
  를 리턴한다. return true; } function writePost() { // 1. 사용자
  입력 값을 가져온다. let contents = $('#contents').val(); // 2. 사
  용자가 올바르게 입력했는지 isValidContents 함수를 통해 확인한다. if
  (isValidContents(contents) == false) { return; } }
```

JavaScript ▼

- ▼ 2: 랜덤으로 유저이름을 생성한다.

```
// 필요한 길이에 맞게 임의의 문자열을 만든다. function
  genRandomName(length) { // 돌려줄 임의의 문자열 let result = '';
  // 임의의 문자열이 될 수 있는 후보 let characters =
  'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let charactersLength = characters.length; // 주어진 길이에 맞게 문
  자열을 만든다. for (let i = 0; i < length; i++) { // 0과 같거나 그
  보다 크고, charactersLength 보다 작은 임의의 숫자를 뽑는다. let
  number = Math.random() * charactersLength; // number 가 소수라면
  정수로 만든다. (ex. 7.23 -> 7) let index = Math.floor(number); //
  index 위치에 있는 문자열 하나를 result에 더한다. result +=
  characters.charAt(index); } // 임의의 문자열을 반환한다. return
  result; } function writePost() { let contents =
  $('#contents').val(); if (isValidContents(contents) === false) {
  return; } // 길이 10의 랜덤 문자열을 뽑는다. let username =
  genRandomName(10); }
```

JavaScript

### ▼ 3: 유저이름과 메모 내용을 서버(/message)에 전달한다.

```
function writePost() { let contents = $('#contents').val(); if
(isValidContents(contents) === false) { return; } let username =
genRandomName(10); $.ajax({ type: "POST", url: "/message", data:
{ 'username_give': username, 'contents_give': contents },
success: function (response) { if (response['result'] ==
'success') { // 서버에서 받아온 메시지를 보여준다.
alert(response['msg']); // 메시지를 저장한 후 창을 새로고침한다. -
> 새롭게 조회하기 위해! window.location.reload(); } } }); }
```

JavaScript

## ▼ (2) 서버 코딩

### ▼ 1: 유저이름과 메모를 받아 DB에 저장하기

```
@app.route('/message', methods=["POST"]) def set_message(): # 클
라이언트가 전달한 username, contents를 변수에 저장한다.
username_receive = request.form['username_give']
contents_receive = request.form['contents_give'] # DB에 저장할
대상을 만든다. doc = { 'username': username_receive, 'contents':
contents_receive, # '지금 시각'을 created_at 이라는 이름으로 함께
저장한다. 'created_at': datetime.now() } # DB에 메모를 저장한다.
db.messages.insert_one(doc) # 성공 메시지를 반환한다. return
jsonify({'result': 'success', 'msg': '메시지 작성에 성공하였습니
다!'})
```

Python

## ▼ (3) 완성코드

### ▼ index.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <met
name="viewport" content="width=device-width, initial-scale=1.0"> <t
코딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap
integrity="sha384-
Vko08x4CGs03+Hhvxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"> <script src="https://code.jquery.com/jquer
3.4.1.slim.min.js" integrity="sha384-
J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
```

```

crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
<script> function isValidContents(contents) { if (contents == '') {
을 입력해주세요'); return false; } if (contents.trim().length > 140) {
백 포함 140자 이하로 입력해주세요'); return false; } return true; }
function getRandomName(length) { let result = ''; let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
charactersLength = characters.length; for (let i = 0; i < length; i++) {
number = Math.random() * charactersLength; let index = Math.floor(r
result += characters.charAt(index); } return result; } function writePost() {
let contents = $('#contents').val(); if (isValidContents(contents)) {
return; } let username = getRandomName(10); $.ajax({ type: "POST",
"/message", data: { 'username_give': username, 'contents_give': contents },
success: function (response) { if (response['result'] == 'success') {
alert(response['msg']); window.location.reload(); } } }); } function openClose() {
let status = $('#post-box').css('display'); if (status == 'block') {
$('#post-box').hide(); } else { $('#post-box').show(); } }
<style> .area-edit { display: none; } .wrap { width: 700px; margin: 10px auto; }
#contents { width: 700px; } .area-write { display: flex; flex-direction: column;
align-items: flex-end; justify-content: center; } .custom-card { margin: 20px 0 20px 0; }
#post-box { display: none; } </style> </head> <body> <div class="wrap"> <div class="header"> <h1>새가 날아든다</h1> <p>
<button onclick="openClose()" type="button" class="btn btn-outline-info btn-sm">
싶은 소식</button>을 입력해주세요. 24시간이 지난 뒤에는 사라집니다. </div>
<div id="post-box"> <div class="area-write"> <textarea placeholder="
은 소식을 입력해주세요" name="contents" id="contents" cols="30" rows="3">
</textarea> <button class="btn btn-danger" onclick="writePost()">작성</button>
</div> </div> <div id="cards-box" class="area-read"> <div class="card"> <header class="card-header"> <p class="card-header-title">
Component </p> <a href="#" class="card-header-icon" aria-label="more">
<span class="icon"> <i class="fas fa-angle-down" aria-hidden="true">
</span> </a> </header> <div class="card-content"> <div class="content">
ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nec in
<a href="#">@bulmaio</a>. <a href="#">#css</a> <a href="#">#response</a>
<time datetime="2016-1-1">11:09 PM - 1 Jan 2016</time> </div> </div>
<div class="card-footer"> <a href="#" class="card-footer-item">Save</a>
<a href="#" class="card-footer-item">Edit</a> <a href="#" class="card-footer-item">Delete</a>
</div> </div> </div> </body> </html>

```



JavaScript ▾

▼ app.py

```

from flask import Flask, render_template, request, jsonify from
pymongo import MongoClient from datetime import datetime,
timedelta app = Flask(__name__) # URL 별로 함수명이 같거나, #
route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return
render_template('index_start.html') @app.route('/message',
methods=["POST"]) def set_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give'] doc = { 'username':
username_receive, 'contents': contents_receive, 'created_at':
datetime.now() } db.messages.insert_one(doc) return
jsonify({'result': 'success', 'msg': '메시지 작성에 성공하였습니
다!'}) if __name__ == '__main__': app.run('0.0.0.0', port=5000,
debug=True)

```

Python ▾

### 3. [새가날아든다] - 최신순으로 메모 정렬하기

## ▼ 5) 최신 순으로 메모를 정렬하고 타임라인에서 조회하기

✅ 필요한 기능은 다음과 같습니다.

**[서버]**

1) 메모 목록을 최신 순으로 정렬해서 돌려주기

**[클라이언트]**

1) 사용자가 접속할 때 메모 목록을 요청하기

2) 전달 받은 메모 목록을 화면에 보여주기

## ▼ (1) 서버 코딩

## ▼ 1: 메모 목록을 최신 순으로 정렬해서 돌려주기

```
@app.route('/message', methods=["GET"]) def get_messages(): #
    created_at 기준 내림차순으로 정렬한다. messages =
    list(db.messages.find({}, {'_id': False}).sort('created_at',
    -1)) # 결과를 messages 라는 key에 저장하여 돌려준다. return
    jsonify({'result': 'success', 'messages': messages})
```

Python ▾

## ▼ (2) 클라이언트 코딩

## ▼ 1: 사용자가 접속할 때 메모 목록을 요청하기

```
// 새로고침할 때마다 getMessages() 함수를 호출한다.
$(document).ready(function () { getMessages(); }) function
getMessages() { // 기존 메모를 삭제한다. $('#cards-box').empty();
// 메모 목록을 요청한다. $.ajax({ type: "GET", url: "/message",
data: {}, success: function (response) { alert('메모를 요청하였습
니다. '); console.log(response['messages']); } }); }
```

JavaScript ▾

## ▼ 2: 전달 받은 메모 목록을 화면에 보여주기

```
$(document).ready(function () { getMessages(); }) function
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET",
url: "/message", data: {}, success: function (response) { if
(response['result'] == 'success') { let messages =
response['messages']; for (let i = 0; i < messages.length; i++)
{ let message = messages[i]; let username = message['username'];
let contents = message['contents']; let created_at =
message['created_at']; // addHTML 함수를 이용해 메모마다 HTML을
```



```

만들고 #cards-box에 붙인다. addHTML(username, contents,
created_at); } } else { alert('메시지를 불러오는데 실패했습니다.');
```

```

}); } // makeMessage 함수를 통해 HTML을 만들고,
#cards-box에 붙인다. function addHTML(username, contents,
created_at) { let tempHtml = makeMessage(username, contents,
created_at); $('#cards-box').append(tempHtml); } // HTML을 만들
고 반환한다. function makeMessage(username, contents,
created_at) { return `<div class="card custom-card"> <div
class="card-body"> <textarea id="${username}-textarea"
class="area-edit" cols="80"></textarea> <h5 id="${username}-
contents" class="card-title">${contents}</h5> <h6
id="${username}-username" class="card-subtitle mb-2 text-
muted">${username}</h6> <h6 class="card-subtitle mb-2 text-
muted">${created_at}</h6> </div> <footer class="card-footer"> <a
id="${username}-edit" href="#" class="card-footer-item"
onclick="editPost('${username}')">수정</a> <a id="${username}-
cancel" href="#" class="card-footer-item area-edit"
onclick="hideEdits('${username}')">취소</a> <a id="${username}-
submit" href="#" class="card-footer-item area-edit"
onclick="submitEdit('${username}')">수정완료</a> </footer>
</div>`; }
```

JavaScript ▾

### ▼ (3) 완성 코드

#### ▼ app.py

```

from flask import Flask, render_template, request, jsonify from
pymongo import MongoClient from datetime import datetime,
timedelta app = Flask(__name__) # URL 별로 함수명이 같거나, #
route('/') 등의 주소가 같으면 안됩니다. client =
MongoClient('localhost', 27017) db = client.bird_comming
@app.route('/') def home(): return render_template('index.html')
@app.route('/message', methods=["GET"]) def get_messages():
messages = list(db.messages.find({}, {'_id':
False})).sort('created_at', -1) return jsonify({'result':
'success', 'messages': messages}) @app.route('/message',
methods=["POST"]) def set_message(): username_receive =
request.form['username_give'] contents_receive =
request.form['contents_give'] doc = { 'username':
username_receive, 'contents': contents_receive, 'created_at':
datetime.now() } db.messages.insert_one(doc) return
jsonify({'result': 'success', 'msg': '메시지 작성에 성공하였습니
다!'}) if __name__ == '__main__': app.run('0.0.0.0', port=5000,
debug=True)
```

JavaScript ▾

#### ▼ index.html

```

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>
코딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGs03+Hhvxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"> <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1lyYfoRJSJoz+n"
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script> $(document).ready(function () { getMessages(); }) function
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET", url: "/api/messages",
data: {}, success: function (response) { if (response['result'] == 'success') {
let messages = response['messages']; for (let i = 0; i < messages.length; i++) {
let message = messages[i]; let username = message['username']; let contents =
message['contents']; let created_at = message['created_at']; addHTML(username,
contents, created_at); } } else { alert('메시지를 불러오는데 실패했습니다'); } }
function addHTML(username, contents, created_at) { let tempHTML = `<div class="card
custom-card"> <div class="card-body"> <textarea id="${username}-edit" class="area-edit"
cols="80"></textarea> <h5 id="${username}-title" class="card-title">${contents}</h5>
<h6 id="${username}-username" class="card-subtitle mb-2 text-muted">${username}</h6>
<h6 class="card-subtitle mb-2 text-muted">${created_at}</h6> </div> <footer class="card-footer">
<a id="${username}-edit" href="#" class="card-footer-item">수정</a>
<a id="${username}-cancel" href="#" class="card-footer-item area-edit">취소</a>
<a id="${username}-submit" href="#" class="card-footer-item area-edit">등록</a>
</footer> </div>`; addHTML(tempHTML); } function makeMessage(username, contents,
created_at) { return `<div class="card custom-card"> <div class="card-body">
<textarea id="${username}-edit" class="area-edit" cols="80"></textarea>
<h5 id="${username}-title" class="card-title">${contents}</h5>
<h6 id="${username}-username" class="card-subtitle mb-2 text-muted">${username}</h6>
<h6 class="card-subtitle mb-2 text-muted">${created_at}</h6> </div>
<footer class="card-footer">
<a id="${username}-edit" href="#" class="card-footer-item">수정</a>
<a id="${username}-cancel" href="#" class="card-footer-item area-edit">취소</a>
<a id="${username}-submit" href="#" class="card-footer-item area-edit">등록</a>
</footer> </div>`; } function isValidContents(contents) { if (contents == '') {
alert('내용을 입력해주세요'); return false; } if (contents.trim().length > 140) {
alert('공백 포함하여 140자 이내로 입력해주세요'); return false; } return true; }
function getRandomName() { let result = ''; let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'; let
charactersLength = characters.length; for (let i = 0; i < length; i++) { let
number = Math.random() * charactersLength; let index = Math.floor(number);
result += characters.charAt(index); } return result; }`

```

JavaScript

#### 4. [새가 날아든다] - 24시간 이내 데이터만 보여주기

## ▼ 6) 24시간이 지난 데이터는 보여주지 않기

✅ 필요한 기능은 다음과 같습니다.

## [서버]

1) 최근 24시간 이내 메모만 가져와서 최신 순으로 정렬 후 반환하기

## ▼ (1) 서버 코딩

```
@app.route('/message', methods=["GET"]) def get_messages(): # 지금
시각 구하기 date_now = datetime.now() # 24시간 전 구하기 date_before
= date_now - timedelta(days=1) messages =
list(db.messages.find({'created_at': {'$gte': date_before, '$lte':
date_now}}, {'_id': False}).sort('created_at', -1)) return
jsonify({'result': 'success', 'messages': messages})
```

Python ▾

## ▼ (2) 완성 코드

```
from flask import Flask, render_template, request, jsonify from
pymongo import MongoClient from datetime import datetime, timedelta
app = Flask(__name__) # URL 별로 함수명이 같거나, # route('/') 등의
주소가 같으면 안됩니다. client = MongoClient('localhost', 27017) db =
client.bird_comming @app.route('/') def home(): return
render_template('index.html') @app.route('/message', methods=
["GET"]) def get_messages(): date_now = datetime.now() date_before =
date_now - timedelta(days=1) messages =
list(db.messages.find({'created_at': { '$gte': date_before, '$lte':
date_now}}, {'_id': False}).sort('created_at', -1)) return
jsonify({'result': 'success', 'messages': messages})
@app.route('/message', methods=["POST"]) def set_message():
username_receive = request.form['username_give'] contents_receive =
request.form['contents_give'] doc = { 'username': username_receive,
'contents': contents_receive, 'created_at': datetime.now() }
db.messages.insert_one(doc) return jsonify({'result': 'success',
'msg': '메시지 작성에 성공하였습니다!'}) if __name__ == '__main__':
app.run('0.0.0.0', port=5000, debug=True)
```

Python ▾

## 5. [새가날아든다] - 메모 수정하고 완성하기

## ▼ 7) 작성한 메모 수정 기능 만들기

✓ 필요한 기능은 다음과 같습니다.

## [클라이언트]

- 1) 수정 버튼을 누르면 (취소, 수정) 버튼이 나타나고, 기존 내용을 입력 칸에 담아두기
- 2) 취소 버튼을 누르면 다시 원래대로 돌아가기
- 3) 수정완료 버튼을 누르면 메모 내용, 유저이름을 서버로 전달하기

## [서버]

- 1) 유저 요청에 따라 메모 내용을 수정하고, 생성 시각을 업데이트하기

## ▼ (1) 클라이언트 코딩

- ▼ 1: 수정 버튼을 누르면 (취소, 수정) 버튼이 나타나고, 기존 내용을 입력 칸에 담아두기

```
// 수정 버튼을 누르면 editPost 함수가 호출된다. // 어떤 메모인지 알기 위해 username 전달 받는다. function editPost(username) { // username이 일치하는 메모에 수정 입력칸을 보여준다. showEdits(username); // 기존 메모 내용을 가져온다. let contents = $('#${username}-contents').text(); // 수정 입력 칸에 기존 메모 내용을 넣어둔다. $('#${username}-textarea').val(contents); } function showEdits(username) { // 수정 입력 칸, 수정완료 버튼, 취소 버튼을 보여준다. $('#${username}-textarea').show(); $('#${username}-submit').show(); $('#${username}-cancel').show(); // 메모 내용, 수정 버튼을 숨긴다. $('#${username}-contents').hide(); $('#${username}-edit').hide(); }
```

JavaScript ▼

- ▼ 2: 취소 버튼을 누르면 다시 원래대로 돌아가기

```
// 취소 버튼을 누르면 hideEdits 함수가 호출된다. // 어떤 메모인지 알기 위해 username 전달 받는다. function hideEdits(username) { // 수정 입력 칸, 수정완료 버튼, 취소 버튼을 숨긴다. $('#${username}-textarea').hide(); $('#${username}-submit').hide(); $('#${username}-cancel').hide(); // 메모 내용, 수정 버튼을 보여준다. $('#${username}-contents').show(); $('#${username}-edit').show(); }
```

JavaScript ▼

- ▼ 3: 수정완료 버튼을 누르면 메모 내용, 유저이름을 서버로 전달하기

```
// 수정완료 버튼을 누르면 submitEdit 함수가 호출된다. // 어떤 메모인지 알기 위해 username 전달 받는다. function
submitEdit(username) { // 수정된 메모 내용을 가져오고 올바른지 검사한다. let contents = $('#${username}-textarea').val(); if
(isValidContents(contents) == false) { return; } // 메모 내용 수정을 서버에 요청한다. $.ajax({ type: "POST", url:
"/message/edit", data: { username_give: username, contents_give: contents }, success: function (response) { if
(response['result'] == 'success') { alert(response['msg']); // 변경된 메모를 보기 위해 창을 새로고침한다.
window.location.reload(); } else { alert('메시지 변경에 실패했습니다. '); } } }); }
```

JavaScript ▾

## ▼ (2) 서버 코딩

## ▼ 1: 유저 요청에 따라 메모 내용을 수정하고, 생성 시각을 업데이트하기

```
@app.route('/message/edit', methods=["POST"]) def
edit_message(): username_receive = request.form['username_give']
contents_receive = request.form['contents_give'] # username 기준으로 메시지를 찾아 내용과 생성 시각을 업데이트합니다.
db.messages.update_one({'username': username_receive}, {'$set': {'contents': contents_receive, 'created_at': datetime.now()}})
return jsonify({'result': 'success', 'msg': '메시지 변경에 성공하였습니다!'})
```

Python ▾

## ▼ (3) 완성 코드

## ▼ index.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>
코딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"> <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script> <script
```

```

src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
<script> $(document).ready(function () { getMessages(); }) function
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET", url:
data: {}, success: function (response) { if (response['result'] ==
let messages = response['messages']; for (let i = 0; i < messages.length; i++)
{ let message = messages[i]; let username = message['username']; let
message['contents']; let created_at = message['created_at']; addHTML(
contents, created_at, i); } } else { alert('메시지를 불러오는데 실패
') } }); } function addHTML(username, contents, created_at, i) { let
makeMessage(username, contents, created_at); $('#cards-box').append(
) function makeMessage(username, contents, created_at, i) { return
`<div class="card custom-card"> <div class="card-body"> <textarea id="${username}-${i}"
class="area-edit" cols="80"></textarea> <h5 id="${username}-${i}"
class="card-title">${contents}</h5> <h6 id="${username}-${i}"
class="card-subtitle">${username}</h6> <h6 class="card-subtitle">
${created_at}</h6> </div> <footer class="card-footer"> <a
id="${username}-${i}" href="#" class="card-footer-item"
onclick="editPost('${username}-${i}')">수정</a> <a id="${username}-${i}"
class="card-footer-item area-edit" onclick="hideEdits('${username}-${i}')"
>수정완료</a> </footer> </div>`;
if (contents.trim().length > 140) { alert('공백 포함
내용을 입력해주세요'); return false; } return true; } function getRandomName()
let result = ''; let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
charactersLength = characters.length; for (let i = 0; i < length; i++)
number = Math.random() * charactersLength; let index = Math.floor(
result += characters.charAt(index); } return result; } function writeMessage()
let contents = $('#contents').val(); if (isValidContents(contents))
return; } let username = getRandomName(10); $.ajax({ type: "POST",
"/message", data: { 'username_give': username, 'contents_give': contents },
success: function (response) { if (response['result'] == 'success')
alert(response['msg']); window.location.reload(); } } }); } function
openClose() { let status = $('#post-box').css('display'); if (status ==
'block') { $('#post-box').hide(); } else { $('#post-box').show(); } }
editPost(username) { showEdits(username); let contents = $('#${username}-
contents').text(); $('#${username}-textarea').val(contents); } function
showEdits(username) { $('#${username}-textarea').show(); $('#${username}-
submit').show(); $('#${username}-cancel').show(); $('#${username}-
contents').hide(); $('#${username}-edit').hide(); } function
hideEdits(username) { $('#${username}-textarea').hide(); $('#${username}-
submit').hide(); $('#${username}-cancel').hide(); $('#${username}-
contents').show(); $('#${username}-edit').show(); } function

```



```

submitEdit(username) { let contents = $('#${username}-textarea').val();
(isValidContents(contents) == false) { return; } $.ajax({ type: "POST",
"/message/edit", data: { username_give: username, contents_give: contents },
success: function (response) { if (response['result'] == 'success') {
alert(response['msg']); window.location.reload(); } else { alert('
실패했습니다. '); } } }); } </script> <style> .area-edit { display: flex;
{ width: 700px; margin: 10px auto; } #contents { width: 700px; } .area-read {
display: flex; flex-direction: column; align-items: flex-end; justify-content:
center; } .custom-card { margin: 20px 0 20px 0; } #post-box { display: flex;
</style> </head> <body> <div class="wrap"> <div class="header"> <h1>새가 날아든다</h1>
<p> <button onclick="openclose()" type="button" class="btn btn-info btn-sm">공유하고 싶은
소식</button>을 입력해주세요. 24시간이 지난 후 다시 입력 가능합니다. </p> </div>
<div id="post-box"> <div class="area-write"> <textarea placeholder="공유하고 싶은
소식을 입력해주세요" name="contents" id="contents" cols="30" rows="10"></textarea>
<button class="btn btn-danger" onclick="writePost()">작성하기</button> </div>
</div> <div id="card"> <div class="area-read"> <div class="card"> <header
class="card-header"> <div class="card-header-title"> Component </div> <a href="#"
class="card-header-link" aria-label="more options"> <span class="icon"> <i class="fas
fa-angle-down" aria-hidden="true"></i> </span> </a> </div> <div class="card-body">
<div class="card-content"> Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Phasellus nec iaculis mauris. <a href="#">@bulmaio</a>. <a href="#">#responsive</a>
<br> <time datetime="2016-1-1">11:09 PM - 2016</time> </div> </div>
<div class="card-footer"> <a href="#">Save</a> <a href="#" class="card-footer-item">Edit</a>
<a href="#" class="card-footer-item">Delete</a> </div> </div> </div> </body>
</html>

```

JavaScript

## ▼ app.py

```

from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient
from datetime import datetime, timedelta

app = Flask(__name__) # URL 별로 함수명이 같거나, # route('/') 등의 주소가 같으면 안됩니다.
client = MongoClient('localhost', 27017)
db = client.bird_comming

@app.route('/') def home(): return render_template('index.html')

@app.route('/message', methods=["GET"]) def get_messages():
    date_now = datetime.now()
    date_before = date_now - timedelta(days=1)
    messages = list(db.messages.find({'created_at': {'$gte': date_before, '$lte': date_now}}, {'_id': False}).sort('created_at', -1))
    return jsonify({'result': 'success', 'messages': messages})

@app.route('/message', methods=["POST"]) def set_message():
    username_receive = request.form['username_give']
    contents_receive = request.form['contents_give']
    doc = {

```



```
'username': username_receive, 'contents': contents_receive,
'created_at': datetime.now() } db.messages.insert_one(doc)
return jsonify({'result': 'success', 'msg': '메시지 작성에 성공하
였습니다!'}) @app.route('/message/edit', methods=["POST"]) def
edit_message(): username_receive = request.form['username_give']
contents_receive = request.form['contents_give']
db.messages.update_one({'username': username_receive}, { '$set':
{'contents': contents_receive, 'created_at': datetime.now()}})
return jsonify({'result': 'success', 'msg': '메시지 변경에 성공하
였습니다!'}) if __name__ == '__main__': app.run('0.0.0.0',
port=5000, debug=True)
```

Python ▾

## ▼ 8) 완성 코드

## ▼ index.html

```

<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>
코딩클럽 | 새가 날아든다</title> <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap
integrity="sha384-
Vko08x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"> <script src="https://code.jquery.com/jquery-
3.4.1.slim.min.js" integrity="sha384-
J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script> <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script> <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.m
integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"
<script> $(document).ready(function () { getMessages(); }) function
getMessages() { $('#cards-box').empty(); $.ajax({ type: "GET", url: "/i
data: {}, success: function (response) { if (response['result'] == 'su
let messages = response['messages']; for (let i = 0; i < messages.leng
{ let message = messages[i]; let username = message['username']; let co
message['contents']; let created_at = message['created_at']; addHTML(u
contents, created_at, i); } } else { alert('메시지를 불러오는데 실패했습
} } }); } function addHTML(username, contents, created_at, i) { let te
makeMessage(username, contents, created_at); $('#cards-box').append(ter
} function makeMessage(username, contents, created_at, i) { return `<d
class="card custom-card"> <div class="card-body"> <textarea id="${user
textarea" class="area-edit" cols="80"></textarea> <h5 id="${username}-
class="card-title">${contents}</h5> <h6 id="${username}-username" class:
subtitle mb-2 text-muted">${username}</h6> <h6 class="card-subtitle mb
muted">${created_at}</h6> </div> <footer class="card-footer"> <a
id="${username}-edit" href="#" class="card-footer-item"
onclick="editPost('${username}')">수정</a> <a id="${username}-cancel"
class="card-footer-item area-edit" onclick="hideEdits('${username}')">
<a id="${username}-submit" href="#" class="card-footer-item area-edit"
onclick="submitEdit('${username}')">수정완료</a> </footer> </div>`; } f
isValidContents(contents) { if (contents == '') { alert('내용을 입력해줘
return false; } if (contents.trim().length > 140) { alert('공백 포함 14
입력해주세요'); return false; } return true; } function getRandomName(1
let result = ''; let characters =

```

```
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'; let
charactersLength = characters.length; for (let i = 0; i < length; i++)
number = Math.random() * charactersLength; let index = Math.floor(numbe
result += characters.charAt(index); } return result; } function writePo
let contents = $('#contents').val(); if (isValidContents(contents) ==
return; } let username = getRandomName(10); $.ajax({ type: "POST", url
"/message", data: { 'username_give': username, 'contents_give': conten
success: function (response) { if (response['result'] == 'success') {
alert(response['msg']); window.location.reload(); } } }); } function
openClose() { let status = $('#post-box').css('display'); if (status ==
'block') { $('#post-box').hide(); } else { $('#post-box').show(); } }
editPost(username) { showEdits(username); let contents = $('#${username}
contents').text(); $('#${username}-textarea').val(contents); } function
showEdits(username) { $('#${username}-textarea').show(); $('#${username}
submit').show(); $('#${username}-cancel').show(); $('#${username}-
contents').hide(); $('#${username}-edit').hide(); } function
hideEdits(username) { $('#${username}-textarea').hide(); $('#${username}
submit').hide(); $('#${username}-cancel').hide(); $('#${username}-
contents').show(); $('#${username}-edit').show(); } function
submitEdit(username) { let contents = $('#${username}-textarea').val()
(isValidContents(contents) == false) { return; } $.ajax({ type: "POST"
"/message/edit", data: { username_give: username, contents_give: conten
success: function (response) { if (response['result'] == 'success') {
alert(response['msg']); window.location.reload(); } else { alert('메시
실패했습니다. '); } } }); } </script> <style> .area-edit { display: none
{ width: 700px; margin: 10px auto; } #contents { width: 700px; } .area
display: flex; flex-direction: column; align-items: flex-end; justify-
center; } .custom-card { margin: 20px 0 20px 0; } #post-box { display:
</style> </head> <body> <div class="wrap"> <div class="header"> <h1>새
다</h1> <p> <button onclick="openClose()" type="button" class="btn btn-
info btn-sm">공유하고 싶은 소식</button>을 입력해주세요. 24시간이 지난 뒤
집니다. </p> </div> <div id="post-box"> <div class="area-write"> <texta
placeholder="공유하고 싶은 소식을 입력해주세요" name="contents" id="cont
cols="30" rows="10"></textarea> <button class="btn btn-danger"
onclick="writePost()">작성하기</button> </div> </div> <div id="cards-bc
class="area-read"> <div class="card"> <header class="card-header"> <p
class="card-header-title"> Component </p> <a href="#" class="card-head
aria-label="more options"> <span class="icon"> <i class="fas fa-angle-r
aria-hidden="true"></i> </span> </a> </header> <div class="card-conten
class="content"> Lorem ipsum dolor sit amet, consectetur adipiscing el
Phasellus nec iaculis mauris. <a href="#">@bulmaio</a>. <a href="#">#c
href="#">#responsive</a> <br> <time datetime="2016-1-1">11:09 PM - 1 J
2016</time> </div> </div> <footer class="card-footer"> <a href="#" cla
footer-item">Save</a> <a href="#" class="card-footer-item">Edit</a> <a
class="card-footer-item">Delete</a> </footer> </div> </div> </div> </b
</html>
```

## ▼ app.py

```

from flask import Flask, render_template, request, jsonify
from pymongo import MongoClient
from datetime import datetime, timedelta

app = Flask(__name__) # URL 별로 함수명이 같거나, # route('/') 등의 주소가 같으면 안됩니다.
client = MongoClient('localhost', 27017)
db = client.bird_comming

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/message', methods=["GET"])
def get_messages():
    date_now = datetime.now()
    date_before = date_now - timedelta(days=1)
    messages = list(db.messages.find({'created_at': {'$gte': date_before, '$lte': date_now}}, {'_id': False}).sort('created_at', -1))
    return jsonify({'result': 'success', 'messages': messages})

@app.route('/message', methods=["POST"])
def set_message():
    username_receive = request.form['username_give']
    contents_receive = request.form['contents_give']
    doc = {'username': username_receive, 'contents': contents_receive, 'created_at': datetime.now()}
    db.messages.insert_one(doc)
    return jsonify({'result': 'success', 'msg': '메시지 작성에 성공하였습니다!'})

@app.route('/message/edit', methods=["POST"])
def edit_message():
    username_receive = request.form['username_give']
    contents_receive = request.form['contents_give']
    db.messages.update_one({'username': username_receive}, {'$set': {'contents': contents_receive, 'created_at': datetime.now()}})
    return jsonify({'result': 'success', 'msg': '메시지 변경에 성공하였습니다!'})

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)

```

Python ▼