



# [스파르타코딩클럽] 웹개발 종합반 - 4주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

## ▼ 단축키 모음

### ▼ 새로고침

- `F5`

### ▼ 저장

- Windows: `Ctrl` + `S`
- macOS: `command` + `S`

### ▼ 전체선택

- Windows: `Ctrl` + `A`
- macOS: `command` + `A`

### ▼ 잘라내기

- Windows: `Ctrl` + `X`
- macOS: `command` + `X`

### ▼ 콘솔창 줄바꿈

- `shift` + `enter`

### ▼ 코드정렬

- Windows: `Ctrl` + `Alt` + `L`
- macOS: `option` + `command` + `L`

### ▼ 들여쓰기

- `Tab`
- 들여쓰기 취소 : `Shift` + `Tab`

### ▼ 주석

- Windows: `Ctrl` + `/`
- macOS: `command` + `/`

## [수업 목표]

1. Flask 프레임워크를 활용해서 API를 만들 수 있다.
2. '화성에 땅사기' API를 만들고 클라이언트에 연결한다.
3. '스파르타피디아' API를 만들고 클라이언트와 연결한다.

## [목차]

- 01. 4주차 오늘 배울 것
- 02. Flask 시작하기 - 서버만들기
- 03. Flask 시작하기 - HTML파일 주기
- 04. Flask시작하기 - 본격 API 만들기
- 05. [화성땅 공동구매] - 프로젝트 세팅
- 06. [화성땅 공동구매] - 뼈대 준비하기
- 07. [화성땅 공동구매] - POST 연습(주문 저장)

08. [화성땅 공동구매] - GET 연습(주문 보여주기)  
09. [스파르타피디아] - 프로젝트 세팅  
10. [스파르타피디아] - 조각 기능 구현해보기  
11. [스파르타피디아] - 뼈대 준비하기  
12. [스파르타피디아] - POST 연습(포스팅하기)  
13. [스파르타피디아] - GET 연습(보여주기)  
14. 4주차 끝 & 숙제 설명  
HW. 4주차 숙제 해설



모든 토글을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

## 01. 4주차 오늘 배울 것

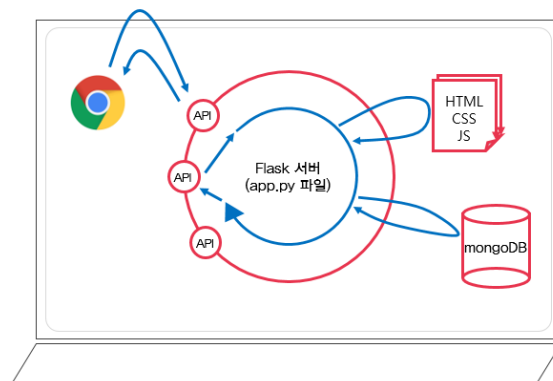
▼ 1) 오늘 배울 것 이야기- 4주차: Flask, 미니프로젝트1, 미니프로젝트2

이번 주 완성본 1. 화성땅공동구매 → [결과물 링크](#)

이번 주 완성본 2. 스파르타피디아 → [결과물 링크](#)



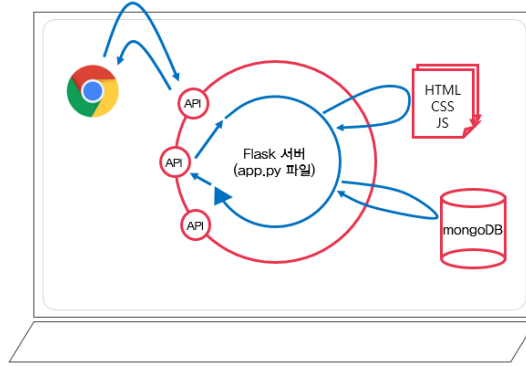
오늘은 HTML과 mongoDB까지 연동해서 서버를 만들어봅시다!



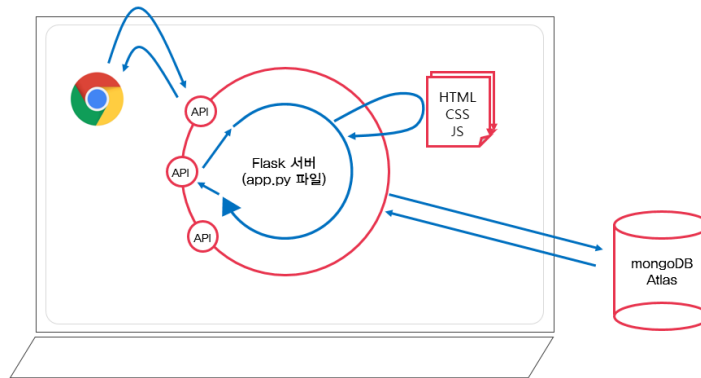
나중에 또 이야기하겠지만 헛갈리면 안되는 것!

우리는 컴퓨터가 한 대 잡아요... 그래서 같은 컴퓨터에다 서버도 만들고, 요청도 할 거예요. 즉, 클라이언트 = 서버가 되는 것이죠.

이것을 바로 "로컬 개발환경"이라고 한답니다! 그림으로 보면, 대략 이렇습니다.



👉 하지만 우리는 **mongoDB Atlas** 라는 클라우드 서비스를 이용하니, 아래와 같겠죠!



▼ 1) **sparta** 폴더 → **projects** 폴더 아래에, 다섯개 만들고 시작하기

👉 웹개발의 꽃, 백엔드-프론트엔드를 연결하는 일이 익숙해지도록,  
연습 → 화성땅공동구매 → 스파르타피디아 → (숙제: 팬명록) → 버킷리스트

총 4번에 걸쳐 반복 실습을 진행 할 예정입니다. 다 해내고 나면 아-주 익숙해질거예요!

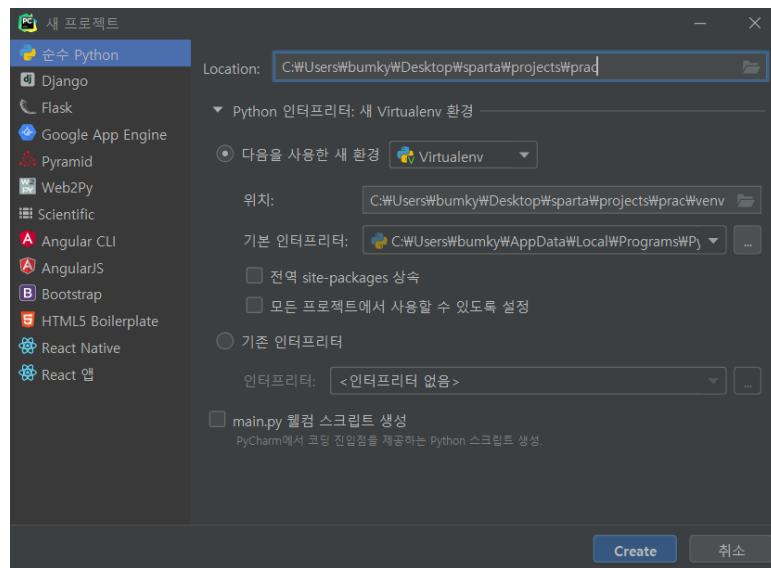
👉 코드 관리를 위해 미리 아래와 같이 폴더구조를 만들고 시작합니다!

- **prac** : "flask 연습 코드를 작성합니다." (오늘)
- **mars** : "화성땅공동구매" 관련 코드를 작성합니다. (오늘)
- **movie** : "스파르타피디아" 관련 코드를 작성합니다. (오늘)
- **homework** : "팬명록" 관련 코드를 작성합니다. (오늘 숙제!)
- **bucket** : "버킷리스트" 관련 코드를 작성합니다. (5주차)

## 02. Flask 시작하기 - 서버만들기

🔥 **sparta** → **projects** → **prac** 폴더에서 시작!

- ▼ 1) new project → prac 폴더를 선택하고 **create** 를 클릭합니다.



- ▼ 2) Flask 패키지 설치하고 시작!

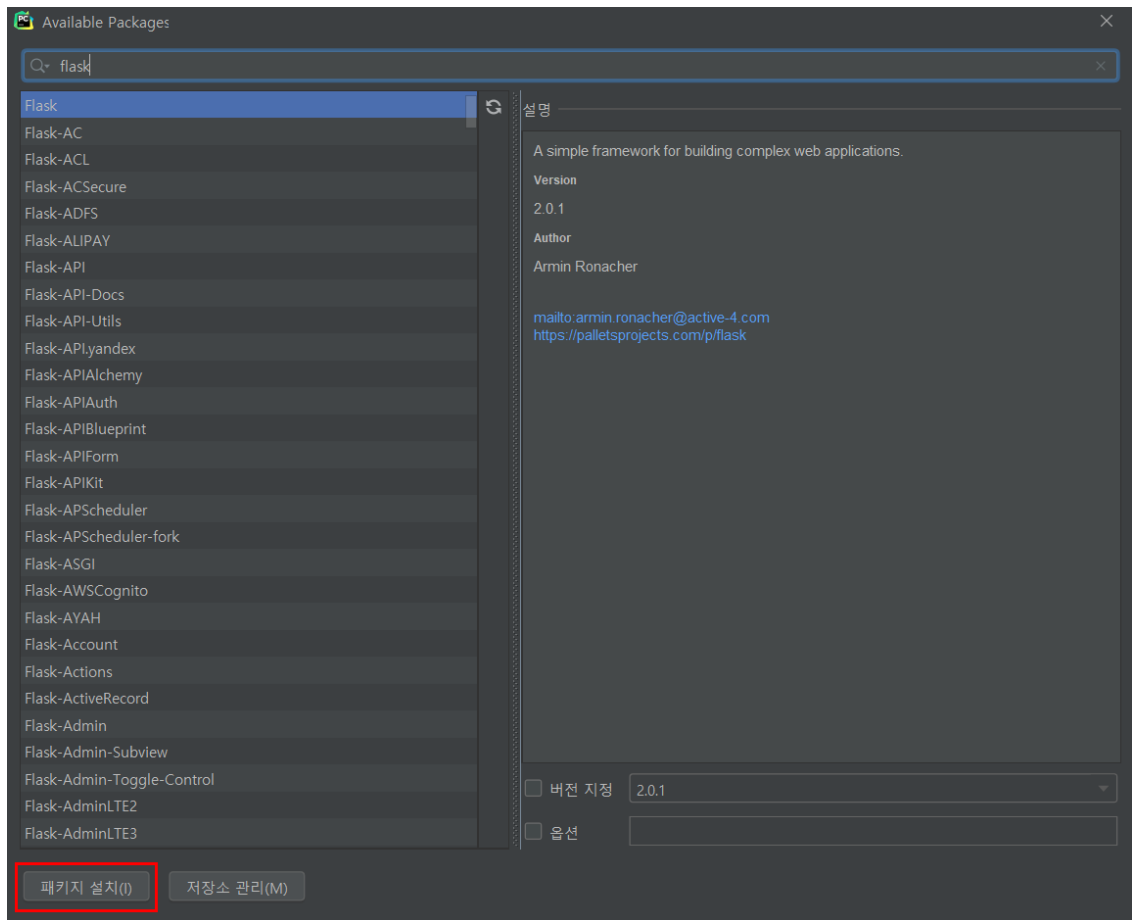


#### 리마인드! 패키지 설치 화면 진입하기

윈도우 : 좌상단File → setting → Python interpreter

맥 : 좌상단Pycharm → Preference → Python Interpreter

- python interpreter 화면에서 + 버튼을 누르면 아래 창이 뜹니다!  
(맥은 아래에, 윈도우는 오른쪽에 위치)
- **flask** 로 검색한 후, Install package 클릭



### ▼ 3) Flask 기초: 기본 실행

- Flask 프레임워크: 서버를 구동시켜주는 편한 코드 모음. 서버를 구동하려면 필요한 복잡한 일들을 쉽게 가져다 쓸 수 있습니다.

👉 프레임워크를 쓰지 않으면 태양초를 빻아서 고추장을 만드는 격!  
프레임워크는 3분 요리/소스 세트라고 생각하면 되겠습니다!

- `app.py` 파일을 만들어 아래 코드를 붙여넣어봅니다.

👉 파일 이름은 아무렇게나 해도 상관없지만, 통상적으로 flask 서버를 돌리는 파일은 `app.py`라고 이름 짓습니다!

### ▼ [코드스니펫] flask 시작 코드

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return 'This is Home!'

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

- 오른쪽 클릭 → 'Run app'을 클릭하고, 터미널에 아래와 같은 메시지가 뜨면 실행 성공!

```
app ×
C:\Users\bumky\Desktop\sparta\projects\prac\venv\Scripts\python.exe C:\Users\bumky\Desktop\sparta\projects\prac\app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 101-376-675
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.53:5000/ (Press CTRL+C to quit)
```

- 이제 크롬에서 <http://localhost:5000/> 으로 접속해보세요.

👉 화면에 `Hello World!` 라는 메시지가 보이시나요? 그렇다면 성공한 것! 🎉

- 종료하는 방법

👉 터미널 창을 클릭하시고, `ctrl + c` 을 누르시면 서버를 종료할 수 있습니다.

#### ▼ 4) Flask 기초: URL 나눠보기

- `@app.route('/')` 부분을 수정해서 URL을 나눌 수 있습니다! 간단하죠?

👉 url 별로 함수명이 같거나, `route('/')`내의 주소가 같으면 안됩니다.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return 'This is Home!'

@app.route('/mypage')
def mypage():
    return 'This is My Page!'

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

## 03. Flask 시작하기 - HTML파일 주기

- ▼ 1) Flask 기초: 기본 폴더구조 - 항상 이렇게 세팅하고 시작!

👉 Flask 서버를 만들 때, 항상,

프로젝트 폴더 안에,  
└ **static** 폴더 (이미지, css파일을 넣어둡니다)  
└ **templates** 폴더 (html파일을 넣어둡니다)  
└ **app.py** 파일

이렇게 세 개를 만들어두고 시작하세요. 이제 각 폴더의 역할을 알아봅시다!

(꼭 참고!! venv는 실제로는 보이지만, **안보인다**~라고 생각하세요! 기억하시죠?)

## ▼ 2) Flask 기초: HTML 파일 불러오기



**templates** 폴더의 역할을 알아보겠습니다.  
HTML 파일을 담아두고, 불러오는 역할을 하죠!

### 1. 간단한 index.html 파일을 templates 안에 만들기

#### ▼ [코드스니펫] index.html 예제코드

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <title>Document</title>

  <script>
    function hey(){
      alert('안녕!')
    }
  </script>
</head>
<body>
  <button onclick="hey()">나는 버튼!</button>
</body>
</html>
```

### 2. html 파일 불러오기



flask 내장함수 `render_template`를 이용합니다. 바로 이게 프레임워크의 위력!

```
from flask import Flask, render_template
app = Flask(__name__)

## URL 별로 함수명이 같거나,
## route('/') 등의 주소가 같으면 안됩니다.

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

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

## 04. Flask시작하기 - 본격 API 만들기

### ▼ 1) 들어가기 전에: GET, POST 요청타입 - 리마인드



**리마인드!**  
은행의 창구가 API와 같다는 것을 기억하시나요?  
같은 예금 창구에서도 개인 고객이나 기업 고객이나에 따라 처리하는 것이 다른 것처럼,

**클라이언트가 요청 할 때에도, "방식"이 존재합니다.**

HTTP 라는 통신 규약을 따른다는 거 잊지 않으셨죠? 클라이언트는 요청할 때 HTTP request method(요청 메소드)를 통해, 어떤 요청 종류인지 응답하는 서버 쪽에 정보를 알려주는 거예요.

## 👉 GET, POST 방식

여러 방식(링크)이 존재하지만 우리는 가장 많이 쓰이는 GET, POST 방식에 대해 다루겠습니다.

- \* GET → 통상적으로! 데이터 조회(Read)를 요청할 때  
예) 영화 목록 조회  
→ 데이터 전달 : URL 뒤에 물음표를 붙여 key=value로 전달  
예: google.com?q=북극곰
- \* POST → 통상적으로! 데이터 생성(Create), 변경(Update), 삭제>Delete) 요청 할 때  
예) 회원가입, 회원탈퇴, 비밀번호 수정  
→ 데이터 전달 : 바로 보이지 않는 HTML body에 key:value 형태로 전달

### ▼ 2) GET, POST 요청에서 클라이언트의 데이터를 받는 방법

- 예를 들어, 클라이언트에서 서버에 title\_give란 키 값으로 데이터를 들고왔다고 생각합시다.  
(주민등록번호 라는 키 값으로 900120- .. 을 가져온 것과 같은 의미)

👉 받은 값을 개발자가 볼 수 있게 print 로 찍어볼 수 있게 했습니다. 실전에선 print로 찍어주는 것 외에, 여러가지 작업을 할 수 있겠죠?

### ▼ [코드스니펫] JQuery 임포트

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

### ▼ [코드스니펫] GET 요청 API코드

```
@app.route('/test', methods=['GET'])
def test_get():
    title_receive = request.args.get('title_give')
    print(title_receive)
    return jsonify({'result':'success', 'msg': '이 요청은 GET!'})
```

### ▼ [코드스니펫] GET 요청 확인 Ajax코드

```
$.ajax({
    type: "GET",
    url: "/test?title_give=봄날은간다",
    data: {},
    success: function(response){
        console.log(response)
    }
})
```

### ▼ [코드스니펫] POST 요청 API코드

```
@app.route('/test', methods=['POST'])
def test_post():
    title_receive = request.form['title_give']
    print(title_receive)
    return jsonify({'result':'success', 'msg': '이 요청은 POST!'})
```

### ▼ [코드스니펫] POST 요청 확인 Ajax코드

```
$.ajax({
    type: "POST",
    url: "/test",
    data: { title_give:'봄날은간다' },
    success: function(response){
        console.log(response)
    }
})
```



```
}  
})
```

## 05. [화성땅 공동구매] - 프로젝트 세팅



sparta → projects → mars 폴더에서 시작!

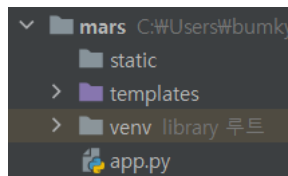
### ▼ 1) 문제 분석 - 완성작부터 보기!

- mars : "화성땅공동구매" 관련 코드를 작성합니다. (오늘)

### ▼ 2) 프로젝트 설정 - flask 폴더 구조 만들기



static, templates 폴더 + app.py 만들기! 이젠 너무 익숙하죠?



### ▼ 3) 패키지 설치하기

- 3개 : flask, pymongo, dnspython

## 06. [화성땅 공동구매] - 뼈대 준비하기

### ▼ 1) 프로젝트 준비 - app.py 준비하기

#### ▼ [코드스니펫] 화성땅 공동구매-app.py

```
from flask import Flask, render_template, request, jsonify  
app = Flask(__name__)  
  
@app.route('/')  
def home():  
    return render_template('index.html')  
  
@app.route("/mars", methods=["POST"])  
def web_mars_post():  
    sample_receive = request.form['sample_give']  
    print(sample_receive)  
    return jsonify({'msg': 'POST 연결 완료!'})  
  
@app.route("/mars", methods=["GET"])  
def web_mars_get():  
    return jsonify({'msg': 'GET 연결 완료!'})  
  
if __name__ == '__main__':  
    app.run('0.0.0.0', port=5000, debug=True)
```

### ▼ 2) 프로젝트 준비 - index.html 준비하기

#### ▼ [코드스니펫] 화성땅 공동구매-index.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztqcQTWfSpd3yD65VohhpuuCOMLASjC" crossorigin="anonymous">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-Mrcw6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/twTiaxVXM"
      crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Gowun+Batang:wght@400;700&display=swap" rel="stylesheet">

<title>선착순 공동구매</title>

<style>
  * {
    font-family: 'Gowun Batang', serif;
    color: white;
  }

  body {
    background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://cdn.aitimes.com/new');
    background-position: center;
    background-size: cover;
  }

  h1 {
    font-weight: bold;
  }

  .order {
    width: 500px;
    margin: 60px auto 0px auto;
    padding-bottom: 60px;
  }

  .mybtn {
    width: 100%;
  }

  .order > table {
    margin: 40px 0;
    font-size: 18px;
  }

  option {
    color: black;
  }
</style>
<script>
  $(document).ready(function () {
    show_order();
  });
  function show_order() {
    $.ajax({
      type: 'GET',
      url: '/mars',
      data: {},
      success: function (response) {
        alert(response['msg'])
      }
    });
  }
  function save_order() {
    $.ajax({
      type: 'POST',
      url: '/mars',
      data: { sample_give: '데이터전송' },
      success: function (response) {
        alert(response['msg'])
      }
    });
  }
</script>
</head>
<body>
  <div class="mask"></div>
  <div class="order">
    <h1>화성에 땅 사내기!</h1>
    <h3>가격: 평 당 500원</h3>
    <p>
      화성에 땅을 사들 수 있다고?<br/>
      앞으로 백년 간 오지 않을 기회. 화성에서 즐기는 노후!
    </p>
    <div class="order-info">
      <div class="input-group mb-3">
        <span class="input-group-text">이름</span>
        <input id="name" type="text" class="form-control">
      </div>
      <div class="input-group mb-3">

```

```

        <span class="input-group-text">주소</span>
        <input id="address" type="text" class="form-control">
    </div>
    <div class="input-group mb-3">
        <label class="input-group-text" for="size">평수</label>
        <select class="form-select" id="size">
            <option selected>-- 주문 평수 --</option>
            <option value="10평">10평</option>
            <option value="20평">20평</option>
            <option value="30평">30평</option>
            <option value="40평">40평</option>
            <option value="50평">50평</option>
        </select>
    </div>
    <button onclick="save_order()" type="button" class="btn btn-warning mybtn">주문하기</button>
</div>
<table class="table">
    <thead>
        <tr>
            <th scope="col">이름</th>
            <th scope="col">주소</th>
            <th scope="col">평수</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>홍길동</td>
            <td>서울시 용산구</td>
            <td>20평</td>
        </tr>
        <tr>
            <td>임꺽정</td>
            <td>부산시 동구</td>
            <td>10평</td>
        </tr>
        <tr>
            <td>세종대왕</td>
            <td>세종시 대왕구</td>
            <td>30평</td>
        </tr>
    </tbody>
</table>
</div>
</body>
</html>

```

### ▼ 3) 프로젝트 준비 - mongoDB Atlas 창 띄워두기

#### ▼ [코드스니펫] mongoDB Atlas 주소

<https://cloud.mongodb.com/>

## 07. [화성땅 공동구매] - POST 연습(주문 저장)

### ▼ 1) API 만들고 사용하기 - 이름, 주소, 평수 저장하기(Create → POST)

- 👉 1. 요청 정보 : URL = /mars, 요청 방식 = POST
- 2. 클라(ajax) → 서버(flask) : name, address, size
- 3. 서버(flask) → 클라(ajax) : 메시지를 보냄 (주문 완료!)

### ▼ 1) 클라이언트와 서버 연결 확인하기

[서버 코드 - app.py]

```

@app.route("/mars", methods=["POST"])
def mars_post():
    sample_receive = request.form['sample_give']
    print(sample_receive)
    return jsonify({'msg': 'POST 연결 완료!'})

```

[클라이언트 코드 - `index.html`]

```
function save_order() {
  $.ajax({
    type: 'POST',
    url: '/mars',
    data: { sample_give:'데이터전송' },
    success: function (response) {
      alert(response['msg'])
    }
  });
}

<button onclick="save_order()" type="button" class="btn btn-warning mybtn">주문하기</button>
```

#### ▼ 2) 서버부터 만들기

- `name`, `address`, `size` 정보를 받아서, 저장하면 되겠죠?
- 우리가 일전에 만들어둔 `dbtest.py` 파일을 불러와봅시다!

```
@app.route("/mars", methods=["POST"])
def mars_post():
    name_receive = request.form['name_give']
    address_receive = request.form['address_give']
    size_receive = request.form['size_give']

    doc = {
        'name': name_receive,
        'address': address_receive,
        'size': size_receive
    }

    db.orders.insert_one(doc)

    return jsonify({'msg': '주문 완료!'})
```

#### ▼ 3) 클라이언트 만들기

- 이번엔 `name`, `address`, `size` 정보를 보내주면 되겠죠?

```
function save_order() {
  let name = $('#name').val()
  let address = $('#address').val()
  let size = $('#size').val()

  $.ajax({
    type: 'POST',
    url: '/mars',
    data: { name_give:name, address_give:address, size_give:size },
    success: function (response) {
      alert(response['msg'])
      window.location.reload()
    }
  });
}
```

#### ▼ 4) 완성 확인하기

- DB에 잘 들어갔는지 확인해보면 되겠죠?



## 08. [화성땅 공동구매] - GET 연습(주문 보여주기)

▼ 1) API 만들고 사용하기 - 저장된 주문을 화면에 보여주기(Read → GET)



1. 요청 정보 : URL= `/mars`, 요청 방식 = `GET`
2. 클라(ajax) → 서버(flask) : (없음)
3. 서버(flask) → 클라(ajax) : 전체 주문을 보내주기

▼ 1) 클라이언트와 서버 확인하기

[서버 코드 - `app.py`]

```
@app.route("/mars", methods=["GET"])
def mars_get():
    return jsonify({'msg': 'GET 연결 완료!'})
```

[클라이언트 코드 - `index.html`]

```
$(document).ready(function () {
    show_order();
});
function show_order() {
    $.ajax({
        type: 'GET',
        url: '/mars',
        data: {},
        success: function (response) {
            alert(response['msg'])
        }
    });
}
```

▼ 2) 서버부터 만들기

- 받을 것 없이 `orders` 에 주문정보를 담아서 내려주기만 하면 됩니다!

```
@app.route("/mars", methods=["GET"])
def mars_get():
    orders_list = list(db.orders.find({}, {'_id': False}))
    return jsonify({'orders': orders_list})
```

▼ 3) 클라이언트 만들기

- 응답을 잘 받아서 for 문으로! 붙여주면 끝이겠조!

```
function show_order() {
    $('#order-box').empty()
    $.ajax({
        type: 'GET',
        url: '/mars',
        data: {},
        success: function (response) {
            let rows = response['orders']
            for (let i = 0; i < rows.length; i++) {
                let name = rows[i]['name']
                let address = rows[i]['address']
                let size = rows[i]['size']

                let temp_html = `<tr>
                    <td>${name}</td>
                    <td>${address}</td>
                    <td>${size}</td>
                </tr>`
                $('#order-box').append(temp_html)
            }
        }
    });
}
```

#### ▼ 4) 완성 확인하기



**동작 테스트:** 화면을 새로고침 했을 때, DB에 저장된 리뷰가 화면에 올바르게 나타나는지 확인합니다.

#### ▼ 2) 전체 완성 코드

##### ▼ [코드 - 서버 `app.py`]

```
from flask import Flask, render_template, request, jsonify
app = Flask(__name__)

from pymongo import MongoClient
client = MongoClient('내 URL')
db = client.dbsparta

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

@app.route("/mars", methods=["POST"])
def mars_post():
    name_receive = request.form['name_give']
    address_receive = request.form['address_give']
    size_receive = request.form['size_give']

    doc = {
        'name': name_receive,
        'address': address_receive,
        'size': size_receive
    }

    db.orders.insert_one(doc)

    return jsonify({'msg': '주문 완료!'})

@app.route("/mars", methods=["GET"])
def mars_get():
    orders_list = list(db.orders.find({}, {'_id': False}))
    return jsonify({'orders': orders_list})

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

##### ▼ [코드 - 클라이언트 `index.html`]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztQCQTWfSpd3yD65VohhpuuCOMLASjC" crossorigin="anonymous">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/twTiaxVXM"
        crossorigin="anonymous"></script>

  <link href="https://fonts.googleapis.com/css2?family=Gowun+Batang:wght@400;700&display=swap" rel="stylesheet">

  <title>선착순 공동구매</title>

  <style>
    * {
      font-family: 'Gowun Batang', serif;
      color: white;
    }

    body {
      background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://cdn.aitimes.com/new');
      background-position: center;
      background-size: cover;
    }

    h1 {
```

```

        font-weight: bold;
    }

    .order {
        width: 500px;
        margin: 60px auto 0px auto;
        padding-bottom: 60px;
    }

    .mybtn {
        width: 100%;
    }

    .order > table {
        margin : 40px 0;
        font-size: 18px;
    }

    option {
        color: black;
    }
}
</style>
<script>
$(document).ready(function () {
    show_order();
});
function show_order() {
    $('#order-box').empty()
    $.ajax({
        type: 'GET',
        url: '/mars',
        data: {},
        success: function (response) {
            let rows = response['orders']
            for (let i = 0; i < rows.length; i++) {
                let name = rows[i]['name']
                let address = rows[i]['address']
                let size = rows[i]['size']

                let temp_html = `<tr>
                    <td>${name}</td>
                    <td>${address}</td>
                    <td>${size}</td>
                </tr>`
                $('#order-box').append(temp_html)
            }
        }
    });
}
function save_order() {
    let name = $('#name').val()
    let address = $('#address').val()
    let size = $('#size').val()

    $.ajax({
        type: 'POST',
        url: '/mars',
        data: { name_give:name, address_give:address, size_give:size },
        success: function (response) {
            alert(response['msg'])
            window.location.reload()
        }
    });
}
}
</script>
</head>
<body>
<div class="mask"></div>
<div class="order">
    <h1>화성에 땅 사내기!</h1>
    <h3>가격: 평 당 500원</h3>
    <p>
        화성에 땅을 사들 수 있다고?<br/>
        앞으로 백년 간 오지 않을 기회. 화성에서 즐기는 노후!
    </p>
    <div class="order-info">
        <div class="input-group mb-3">
            <span class="input-group-text">이름</span>
            <input id="name" type="text" class="form-control">
        </div>
        <div class="input-group mb-3">
            <span class="input-group-text">주소</span>
            <input id="address" type="text" class="form-control">
        </div>
        <div class="input-group mb-3">
            <label class="input-group-text" for="size">평수</label>


```

```

        <select class="form-select" id="size">
            <option selected>-- 주문 평수 --</option>
            <option value="10평">10평</option>
            <option value="20평">20평</option>
            <option value="30평">30평</option>
            <option value="40평">40평</option>
            <option value="50평">50평</option>
        </select>
    </div>
    <button onclick="save_order()" type="button" class="btn btn-warning mybtn">주문하기</button>
</div>
<table class="table">
    <thead>
        <tr>
            <th scope="col">이름</th>
            <th scope="col">주소</th>
            <th scope="col">평수</th>
        </tr>
    </thead>
    <tbody id="order-box">
        <tr>
            <td>홍길동</td>
            <td>서울시 용산구</td>
            <td>20평</td>
        </tr>
        <tr>
            <td>임꺽정</td>
            <td>부산시 동구</td>
            <td>10평</td>
        </tr>
        <tr>
            <td>세종대왕</td>
            <td>세종시 대왕구</td>
            <td>30평</td>
        </tr>
    </tbody>
</table>
</div>
</body>
</html>

```

## 09. [스파르타피디아] - 프로젝트 세팅

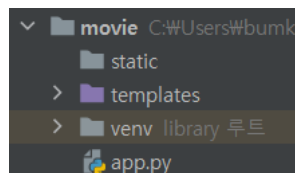
 sparta → projects → movie 폴더에서 시작!

### ▼ 1) 문제 분석 - 완성작부터 보기!

- **movie** : "스파르타피디아" 관련 코드를 작성합니다. (오늘)

### ▼ 2) 프로젝트 준비 - flask 폴더 구조 만들기

 static, templates 폴더 + **app.py** 만들기! 이젠 너무 익숙하죠?



### ▼ 3) 패키지 설치하기

- 5개 : flask, pymongo, dnspython, bs4, requests

## 10. [스파르타피디아] - 조각 기능 구현해보기

### ▼ 1) 프로젝트 준비 - URL에서 페이지 정보 가져오기 (meta태그 스크래핑)



👉 이렇게, API에서 수행해야하는 작업 중 익숙하지 않은 것들은, 따로 python 파일을 만들어 실행해보고, 잘 되면 코드를 붙여넣는 방식으로 하는 게 편합니다.

그럼, meta tag가 뭔지 공부해볼까요?

▼ 어떤 부분에 스크래핑이 필요한가요?

- 우리는 URL만 입력했는데, 자동으로 불러와지는 부분들이 있습니다.

👉 함께 확인해볼까요? [스파르타피디아](#)

- 바로 '제목', '썸네일 이미지', '내용' 입니다.

👉 이 부분은, 'meta'태그를 크롤링 함으로써 공통적으로 얻을 수 있습니다.  
meta태그가 무엇이고, 어떻게 스크래핑 하는지, 함께 살펴볼까요?



보스 베이비 2

베이비 주식회사의 레전드 보스 베이비에서 인생 만렙 CEO가 된 '테드' 베이비인 줄 알았던 조카 '티...

★★★★★

가볍게 보러갔다가 감동까지 느끼고  
웁니다올해 봤던 애니메이션 중에 제일  
재밌게 봤네요



더 수어사이드 스쿼드

살고 싶다면 무조건 성공시켜라! 최강  
우주 빌런에 맞선, 자살특공대에게 맡겨진 '더' 대책 없는 작전...

★★★★★

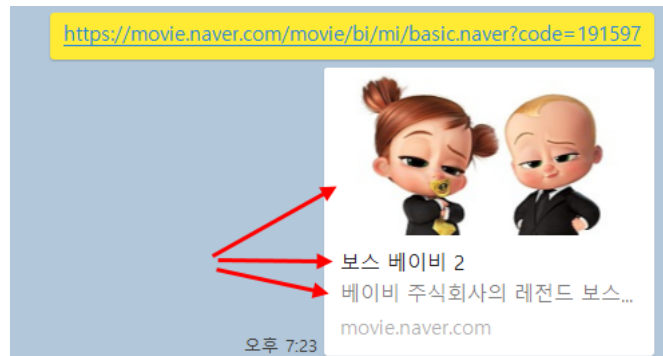
사방에 또라이들 천지네 ㅋㅋ

▼ meta 태그에 대해 알아보기

- (링크)에 접속한 뒤 크롬 개발자 도구를 이용해 HTML의 생김새를 살펴볼까요?
- 메타 태그는, <head></head> 부분에 들어가는, 눈으로 보이는 것(body) 외에 사이트의 속성을 설명해주는 태그들입니다.

예) 구글 검색 시 표시 될 설명문, 사이트 제목, 카톡 공유 시 표시 될 이미지 등

- 우리는 그 중 og:image / og:title / og:description 을 크롤링 할 예정입니다.



#### ▼ meta 태그 스크래핑 하기

- 연습을 위해 meta\_prac.py 파일을 만들어봅니다. 기본 준비를 합니다.

##### ▼ [코드스니펫] - 크롤링 기본 코드

```
import requests
from bs4 import BeautifulSoup

url = 'https://movie.naver.com/movie/bi/mi/basic.naver?code=191597'

headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3671.64 Safari/537.36'}
data = requests.get(url,headers=headers)

soup = BeautifulSoup(data.text, 'html.parser')

# 여기에 코딩을 해서 meta tag를 먼저 가져와보겠습니다.
```

```
import requests
from bs4 import BeautifulSoup

url = 'https://movie.naver.com/movie/bi/mi/basic.naver?code=191597'

headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3671.64 Safari/537.36'}
data = requests.get(url,headers=headers)

soup = BeautifulSoup(data.text, 'html.parser')

# 여기에 코딩을 해서 meta tag를 먼저 가져와보겠습니다.
```

- select\_one을 이용해 meta tag를 먼저 가져와봅니다.

👉 select의 새로운 사용법! 이렇게 또 알아가네요~!

```
og_image = soup.select_one('meta[property="og:image"]')
og_title = soup.select_one('meta[property="og:title"]')
og_description = soup.select_one('meta[property="og:description"]')

print(og_image)
print(og_title)
print(og_description)
```

- 가져온 meta tag의 content를 가져와봅시다.

```
image = og_image['content']
title = og_title['content']
description = og_description['content']

print(image)
```

```
print(title)
print(description)
```

## 11. [스파르타피디아] - 뼈대 준비하기

- ### ▼ 1) 프로젝트 준비 - `app.py`, `index.html` 준비하기

▼ [코드스니펫] 스파르타피디아-app.py

```
from flask import Flask, render_template, request, jsonify
app = Flask(__name__)

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

@app.route("/movie", methods=["POST"])
def movie_post():
    sample_receive = request.form['sample_give']
    print(sample_receive)
    return jsonify({'msg': 'POST 연결 완료!'})

@app.route("/movie", methods=["GET"])
def movie_get():
    return jsonify({'msg': 'GET 연결 완료!'})

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

## ▼ [코드스니펫] 스파르타피디아-index.html

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspd3yD65VohhpUuC0mLASjC" crossorigin="anonymous">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
        crossorigin="anonymous"></script>

    <title>스파르타 피디아</title>

    <link href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&display=swap" rel="stylesheet">

    <style>
        * {
            font-family: 'Gowun Dodum', sans-serif;
        }

        .mytitle {
            width: 100%;
            height: 250px;

            background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://movie-phinf.pstatic.net');
            background-position: center;
            background-size: cover;

            color: white;

            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
        }

        .mytitle > button {
            width: 200px;
            height: 50px;

            background-color: transparent;
            color: white;

            border-radius: 50px;
        }
    </style>

```

```

        border: 1px solid white;

        margin-top: 10px;
    }

    .mytitle > button:hover {
        border: 2px solid white;
    }

    .mycomment {
        color: gray;
    }

    .mycards {
        margin: 20px auto 0px auto;
        width: 95%;
        max-width: 1200px;
    }

    .mypost {
        width: 95%;
        max-width: 500px;
        margin: 20px auto 0px auto;
        padding: 20px;
        box-shadow: 0px 0px 3px 0px gray;

        display: none;
    }

    .mybtns {
        display: flex;
        flex-direction: row;
        align-items: center;
        justify-content: center;

        margin-top: 20px;
    }
    .mybtns > button {
        margin-right: 10px;
    }
</style>
<script>
    $(document).ready(function(){
        listing();
    });

    function listing() {
        $.ajax({
            type: 'GET',
            url: '/movie',
            data: {},
            success: function (response) {
                alert(response['msg'])
            }
        })
    }

    function posting() {
        $.ajax({
            type: 'POST',
            url: '/movie',
            data: {sample_give: '데이터전송'},
            success: function (response) {
                alert(response['msg'])
            }
        })
    }

    function open_box(){
        $('#post-box').show()
    }
    function close_box(){
        $('#post-box').hide()
    }
</script>
</head>

<body>
<div class="mytitle">
    <h1>내 생애 최고의 영화들</h1>
    <button onclick="open_box()">영화 기록하기</button>
</div>
<div class="mypost" id="post-box">
    <div class="form-floating mb-3">
        <input id="url" type="email" class="form-control" placeholder="name@example.com">
        <label>영화URL</label>
    </div>

```

```

<div class="input-group mb-3">
  <label class="input-group-text" for="inputGroupSelect01">별점</label>
  <select class="form-select" id="star">
    <option selected>-- 선택하기 --</option>
    <option value="1">★</option>
    <option value="2">★★</option>
    <option value="3">★★★</option>
    <option value="4">★★★★</option>
    <option value="5">★★★★★</option>
  </select>
</div>
<div class="form-floating">
  <textarea id="comment" class="form-control" placeholder="Leave a comment here"></textarea>
  <label for="floatingTextarea2">코멘트</label>
</div>
<div class="mybtns">
  <button onclick="posting()" type="button" class="btn btn-dark">기록하기</button>
  <button onclick="close_box()" type="button" class="btn btn-outline-dark">닫기</button>
</div>
</div>
<div class="mycards">
  <div class="row row-cols-1 row-cols-md-4 g-4" id="cards-box">
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>★★★</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>★★★</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>★★★</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>★★★</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

## 12. [스파르타피디아] - POST 연습(포스팅하기)

▼ 2) API 만들고 사용하기 - 포스팅API (Create → POST)



1. 요청 정보 : URL = `/movie`, 요청 방식 = `POST`
2. 클라(ajax) → 서버(flask) : `url`, `star`, `comment`
3. 서버(flask) → 클라(ajax) : 메시지를 보냄 (포스팅 완료!)

### ▼ 1) 클라이언트와 서버 연결 확인하기

[서버 코드 - `app.py`]

```
@app.route("/movie", methods=["POST"])
def movie_post():
    sample_receive = request.form['sample_give']
    print(sample_receive)
    return jsonify({'msg': 'POST 연결 완료!'})
```

[클라이언트 코드 - `index.html`]

```
function posting() {
    $.ajax({
        type: 'POST',
        url: '/movie',
        data: {sample_give: '데이터전송'},
        success: function (response) {
            alert(response['msg'])
        }
    });
}

<button onclick="posting()" type="button" class="btn btn-dark">기록하기</button>
```

### ▼ 2) 서버부터 만들기

- `url`, `star`, `comment` 정보를 받아서, 저장하면 되겠죠?
- 그리고, 우리가 미리 만든 `meta_prac.py` 도 참고해서 붙여봅시다!
- 우리가 일전에 만들어둔 `dbtest.py` 파일을 불러와봅시다!

```
@app.route("/movie", methods=["POST"])
def movie_post():
    url_receive = request.form['url_give']
    star_receive = request.form['star_give']
    comment_receive = request.form['comment_give']

    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Sa
    data = requests.get(url_receive, headers=headers)

    soup = BeautifulSoup(data.text, 'html.parser')

    og_image = soup.select_one('meta[property="og:image"]')
    og_title = soup.select_one('meta[property="og:title"]')
    og_description = soup.select_one('meta[property="og:description"]')

    image = og_image['content']
    title = og_title['content']
    description = og_description['content']

    doc = {
        'image': image,
        'title': title,
        'desc': description,
        'star': star_receive,
        'comment': comment_receive
    }

    db.movies.insert_one(doc)

    return jsonify({'msg': 'POST 연결 완료!'})
```

### ▼ 3) 클라이언트 만들기

- 이번엔 `url`, `star`, `comment` 정보를 보내주면 되겠죠?

```
function posting() {
  let url = $('#url').val()
  let star = $('#star').val()
  let comment = $('#comment').val()

  $.ajax({
    type: 'POST',
    url: '/movie',
    data: {url_give: url, star_give: star, comment_give: comment},
    success: function (response) {
      alert(response['msg'])
      window.location.reload()
    }
  });
}
```

#### ▼ 4) 완성 확인하기

- DB에 잘 들어갔는지 확인해보면 되겠죠?
- 그 전에, 연습했던 `movies` 컬렉션을 삭제해줄게요!

**dbsparta.movies**

COLLECTION SIZE: 381B   TOTAL DOCUMENTS: 1   INDEXES TOTAL SIZE: 20KB

**Find**   Indexes   Schema Anti-Patterns   Aggregation   Search Indexes ●

**FILTER** { field: 'value' }

**QUERY RESULTS 1-1 OF 1**

```
{
  "_id": "ObjectId(\"612cb955daeb1adb5f2ab562\")",
  "image": "https://movie-phinf.pstatic.net/20210622_174/1624324910624JhEq2_JPEG/m...",
  "title": "보스 베이비 2",
  "desc": "베이비 주식회사의 레전드 보스 베이비에서 인생 만렙 CEO가 된 '데드' 베이비인 줄 알았던 조카 '티...'",
  "star": "5",
  "comment": "그렇게 재밌었는데 보고 싶다!"
}
```

## 13. [스파르타피디아] - GET 연습(보여주기)

### ▼ 1) API 만들고 사용하기 - 보여주기API (Read → GET)

- 👉 1. 요청 정보 : URL = `/movie`, 요청 방식 = `GET`
2. 클라(ajax) → 서버(flask) : (없음)
3. 서버(flask) → 클라(ajax) : 전체 영화를 보내주기

#### ▼ 1) 클라이언트와 서버 확인하기

[서버 코드 - `app.py`]

```
@app.route("/movie", methods=["GET"])
def movie_get():
    return jsonify({'msg': 'GET 연결 완료!'})
```

[클라이언트 코드 - `index.html`]

```
$(document).ready(function(){
    listing();
});

function listing() {
    $.ajax({
        type: 'GET',
        url: '/movie',
        data: {},
        success: function (response) {
            alert(response['msg'])
        }
    })
}
```

## ▼ 2) 서버부터 만들기

- 받을 것 없이 `orders` 에 주문정보를 담아서 내려주기만 하면 됩니다!

```
@app.route("/movie", methods=['GET'])
def movie_get():
    movie_list = list(db.movies.find({}, {'_id': False}))
    return jsonify({'movies': movie_list})
```

## ▼ 3) 클라이언트 만들기

- 응답을 잘 받아서 for 문으로! 붙여주면 끝이겠죠!

```
function listing() {
    $('#cards-box').empty()
    $.ajax({
        type: 'GET',
        url: '/movie',
        data: {},
        success: function (response) {
            let rows = response['movies']
            for(let i = 0; i < rows.length; i++) {
                let image = rows[i]['image']
                let title = rows[i]['title']
                let desc = rows[i]['desc']
                let star = rows[i]['star']
                let comment = rows[i]['comment']

                let star_image = '☆'.repeat(star)

                let temp_html = `<div class="col">
                    <div class="card h-100">
                        
                        <div class="card-body">
                            <h5 class="card-title">${title}</h5>
                            <p class="card-text">${desc}</p>
                            <p>${star_image}</p>
                            <p class="mycomment">${comment}</p>
                        </div>
                    </div>
                </div>`

                $('#cards-box').append(temp_html)
            }
        }
    })
}
```

## ▼ 4) 완성 확인하기



**동작 테스트:** 화면을 새로고침 했을 때, DB에 저장된 리뷰가 화면에 올바르게 나타나는지 확인합니다.

## ▼ 2) 전체 완성 코드

- ▼ 클라이언트 코드 `index.html`



```

<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxvXM"
        crossorigin="anonymous"></script>

  <title>스파르타 피디아</title>

  <link href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&display=swap" rel="stylesheet">

  <style>
    * {
      font-family: 'Gowun Dodum', sans-serif;
    }

    .mytitle {
      width: 100%;
      height: 250px;

      background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://movie-phinf.pstatic.com/');
      background-position: center;
      background-size: cover;

      color: white;

      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
    }

    .mytitle > button {
      width: 200px;
      height: 50px;

      background-color: transparent;
      color: white;

      border-radius: 50px;
      border: 1px solid white;

      margin-top: 10px;
    }

    .mytitle > button:hover {
      border: 2px solid white;
    }

    .mycomment {
      color: gray;
    }

    .mycards {
      margin: 20px auto 0px auto;
      width: 95%;
      max-width: 1200px;
    }

    .mypost {
      width: 95%;
      max-width: 500px;
      margin: 20px auto 0px auto;
      padding: 20px;
      box-shadow: 0px 0px 3px 0px gray;

      display: none;
    }

    .mybtns {
      display: flex;
      flex-direction: row;
      align-items: center;
      justify-content: center;

      margin-top: 20px;
    }

    .mybtns > button {

```

```

        margin-right: 10px;
    }
</style>
<script>
    $(document).ready(function(){
        listing();
    });

    function listing() {
        $('#cards-box').empty()
        $.ajax({
            type: 'GET',
            url: '/movie',
            data: {},
            success: function (response) {
                let rows = response['movies']
                for(let i = 0; i < rows.length; i++) {
                    let image = rows[i]['image']
                    let title = rows[i]['title']
                    let desc = rows[i]['desc']
                    let star = rows[i]['star']
                    let comment = rows[i]['comment']

                    let star_image = '*'.repeat(star)

                    let temp_html = `<div class="col">
                        <div class="card h-100">
                            
                            <div class="card-body">
                                <h5 class="card-title">${title}</h5>
                                <p class="card-text">${desc}</p>
                                <p>${star_image}</p>
                                <p class="mycomment">${comment}</p>
                            </div>
                        </div>
                    </div>`

                    $('#cards-box').append(temp_html)
                }
            }
        })
    }

    function posting() {
        let url = $('#url').val()
        let star = $('#star').val()
        let comment = $('#comment').val()

        $.ajax({
            type: 'POST',
            url: '/movie',
            data: {url_give: url, star_give: star, comment_give: comment},
            success: function (response) {
                alert(response['msg'])
                window.location.reload()
            }
        })
    }

    function open_box(){
        $('#post-box').show()
    }
    function close_box(){
        $('#post-box').hide()
    }
</script>
</head>

<body>
<div class="mytitle">
    <h1>내 생애 최고의 영화들</h1>
    <button onclick="open_box()">영화 기록하기</button>
</div>
<div class="mypost" id="post-box">
    <div class="form-floating mb-3">
        <input id="url" type="email" class="form-control" placeholder="name@example.com">
        <label>영화URL</label>
    </div>
    <div class="input-group mb-3">
        <label class="input-group-text" for="inputGroupSelect01">별점</label>
        <select class="form-select" id="star">
            <option selected>-- 선택하기 --</option>
            <option value="1">★</option>
            <option value="2">★★</option>
            <option value="3">★★★</option>
            <option value="4">★★★★</option>
        </select>
    </div>

```

```

<option value="5">*****</option>
</select>
</div>
<div class="form-floating">
  <textarea id="comment" class="form-control" placeholder="Leave a comment here"></textarea>
  <label for="floatingTextarea2">코멘트</label>
</div>
<div class="mybtns">
  <button onclick="posting()" type="button" class="btn btn-dark">기록하기</button>
  <button onclick="close_box()" type="button" class="btn btn-outline-dark">닫기</button>
</div>
</div>
<div class="mycards">
  <div class="row row-cols-1 row-cols-md-4 g-4" id="cards-box">
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

▼ 서버 코드 `app.py`

```
from flask import Flask, render_template, request, jsonify
app = Flask(__name__)

from pymongo import MongoClient
client = MongoClient('4 URL')
db = client.dbsparta

import requests
from bs4 import BeautifulSoup

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

```

@app.route("/movie", methods=["POST"])
def movie_post():
    url_receive = request.form['url_give']
    star_receive = request.form['star_give']
    comment_receive = request.form['comment_give']

    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Sa
    data = requests.get(url_receive, headers=headers)

    soup = BeautifulSoup(data.text, 'html.parser')

    og_image = soup.select_one('meta[property="og:image"]')
    og_title = soup.select_one('meta[property="og:title"]')
    og_description = soup.select_one('meta[property="og:description"]')

    image = og_image['content']
    title = og_title['content']
    description = og_description['content']

    doc = {
        'image':image,
        'title':title,
        'desc':description,
        'star':star_receive,
        'comment':comment_receive
    }

    db.movies.insert_one(doc)

    return jsonify({'msg':'POST 연결 완료!'})

@app.route("/movie", methods=["GET"])
def movie_get():
    movies_list = list(db.movies.find({}, {'_id':False}))
    return jsonify({'movies':movies_list})

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

```

## 14. 4주차 끝 & 숙제 설명



**1주차에 완성한 팬명록 을 완성해주세요!**

두 가지 기능을 수행해야 합니다.

- 1) 응원 남기기(POST): 정보 입력 후 '응원 남기기' 버튼클릭 시 주문목록에 추가
- 2) 응원 보기(GET): 페이지 로딩 후 하단 응원 목록이 자동으로 보이기

아래 완성본을 참고해주세요!

<http://spartacodingclub.shop/web/homework>

### ▼ [코드스니펫] - 팬명록 보러가기

<http://spartacodingclub.shop/web/homework>

### ▼ [코드 app.py] - 백대 코드로 사용하세요!

#### ▼ [코드스니펫] 팬명록-app.py

```

from flask import Flask, render_template, request, jsonify
app = Flask(__name__)

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

@app.route("/homework", methods=["POST"])
def homework_post():
    sample_receive = request.form['sample_give']

```

```
print(sample_receive)
return jsonify({'msg': 'POST 연결 완료!'})

@app.route("/homework", methods=["GET"])
def homework_get():
    return jsonify({'msg': 'GET 연결 완료!'})

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

▼ [💻 코드 `index.html`] - 뼈대 코드를 사용하세요! (아티스트만 바꾸기!)

▼ [코드스니펫] 팬명록-index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpUuCOMLASjC" crossorigin="anonymous">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
    crossorigin="anonymous"></script>

<title>초미니홈피 - 팬명록</title>

<link href="https://fonts.googleapis.com/css2?family=Noto+Serif+KR:wght@200;300;400;500;600;700;900&display=swap" rel="stylesheet">
<style>
  * {
    font-family: 'Noto Serif KR', serif;
  }
  .mypic {
    width: 100%;
    height: 300px;

    background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('아티스트 사진URL');
    background-position: center 30%;
    background-size: cover;

    color: white;

    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
  }
  .mypost {
    width: 95%;
    max-width: 500px;
    margin: 20px auto 20px auto;

    box-shadow: 0px 0px 3px 0px black;
    padding: 20px;
  }
  .mypost > button {
    margin-top: 15px;
  }
  .mycards {
    width: 95%;
    max-width: 500px;
    margin: auto;
  }
  .mycards > .card {
    margin-top: 10px;
    margin-bottom: 10px;
  }
</style>
<script>
  $(document).ready(function(){
    set_temp()
    show_comment()
  });
  function set_temp(){
    $.ajax({
      type: "GET",
      url: "http://spartacodingclub.shop/sparta_api/weather/seoul",
    })
  }
</script>
```

```

        data: {},
        success: function (response) {
            $('#temp').text(response['temp'])
        }
    })
}
function save_comment(){
    $.ajax({
        type: 'POST',
        url: '/homework',
        data: {sample_give:'데이터전송'},
        success: function (response) {
            alert(response['msg'])
            window.location.reload()
        }
    })
}
function show_comment(){
    $.ajax({
        type: "GET",
        url: "/homework",
        data: {},
        success: function (response) {
            alert(response["msg"])
        }
    });
}
</script>
</head>
<body>
<div class="mypic">
<h1>아티스트이름 팬명록</h1>
<p>현재기온: <span id="temp">36</span>도</p>
</div>
<div class="mypost">
<div class="form-floating mb-3">
<input type="text" class="form-control" id="name" placeholder="url">
<label for="floatingInput">닉네임</label>
</div>
<div class="form-floating">
<textarea class="form-control" placeholder="Leave a comment here" id="comment"
style="height: 100px"></textarea>
<label for="floatingTextarea2">응원댓글</label>
</div>
<button onclick="save_comment()" type="button" class="btn btn-dark">응원 남기기</button>
</div>
<div class="mycards" id="comment-list">
<div class="card">
<div class="card-body">
<blockquote class="blockquote mb-0">
<p>새로운 앨범 너무 멋져요!</p>
<footer class="blockquote-footer">호행맨</footer>
</blockquote>
</div>
</div>
<div class="card">
<div class="card-body">
<blockquote class="blockquote mb-0">
<p>새로운 앨범 너무 멋져요!</p>
<footer class="blockquote-footer">호행맨</footer>
</blockquote>
</div>
</div>
<div class="card">
<div class="card-body">
<blockquote class="blockquote mb-0">
<p>새로운 앨범 너무 멋져요!</p>
<footer class="blockquote-footer">호행맨</footer>
</blockquote>
</div>
</div>
</div>
</body>
</html>

```

## HW. 4주차 숙제 해설

## ▼ [코드스니펫] app.py

```
from flask import Flask, render_template, request, jsonify
app = Flask(__name__)
```

```

from pymongo import MongoClient
client = MongoClient("내 URL")
db = client.sparta

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

@app.route("/homework", methods=["POST"])
def homework_post():
    name_receive = request.form["name_give"]
    comment_receive = request.form["comment_give"]

    doc = {
        'name': name_receive,
        'comment': comment_receive
    }

    db.homework.insert_one(doc)
    return jsonify({'msg': '응원 완료!'})

@app.route("/homework", methods=["GET"])
def homework_get():
    comment_list = list(db.homework.find({}, {'_id': False}))
    return jsonify({'comments': comment_list})

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

```

#### ▼ [코드스니펫] index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztCQTWfspd3yD65VohhpUuCOMLASjC" crossorigin="anonymous">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
        crossorigin="anonymous"></script>

    <title>초미니홈피 - 팬명록</title>

    <link href="https://fonts.googleapis.com/css2?family=Noto+Serif+KR:wght@200;300;400;500;600;700;900&display=swap" rel="stylesheet"
    <style>
        * {
            font-family: 'Noto Serif KR', serif;
        }
        .mypic {
            width: 100%;
            height: 300px;

            background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://cdn.topstarnews.net/news
            background-position: center 30%;
            background-size: cover;

            color: white;

            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
        }

        .mypost {
            width: 95%;
            max-width: 500px;
            margin: 20px auto 20px auto;

            box-shadow: 0px 0px 3px 0px black;
            padding: 20px;
        }

        .mypost > button {
            margin-top: 15px;
        }

        .mycards {

```

```

        width: 95%;
        max-width: 500px;
        margin: auto;
    }

    .mycards > .card {
        margin-top: 10px;
        margin-bottom: 10px;
    }
}
</style>
<script>
    $(document).ready(function () {
        show_comment();
    });
    function save_comment() {
        let name = $('#name').val()
        let comment = $('#comment').val()

        $.ajax({
            type: "POST",
            url: "/homework",
            data: {'name_give':name, 'comment_give':comment},
            success: function (response) {
                alert(response["msg"])
                window.location.reload()
            }
        });
    }
    function show_comment() {
        $('#comment-list').empty()
        $.ajax({
            type: "GET",
            url: "/homework",
            data: {},
            success: function (response) {
                let rows = response['comments']
                for (let i = 0; i < rows.length; i++) {
                    let name = rows[i]['name']
                    let comment = rows[i]['comment']

                    let temp_html = `<div class="card">
                        <div class="card-body">
                            <blockquote class="blockquote mb-0">
                                <p>${comment}</p>
                                <footer class="blockquote-footer">${name}</footer>
                            </blockquote>
                        </div>
                    </div>`
                    $('#comment-list').append(temp_html)
                }
            }
        });
    }
}
</script>
</head>
<body>
    <div class="mypic">
        <h1>심센치(10cm) 팬명록</h1>
        <p>현재개은: <span id="temp">36</span>도</p>
    </div>
    <div class="mypost">
        <div class="form-floating mb-3">
            <input type="text" class="form-control" id="name" placeholder="url">
            <label for="floatingInput">닉네임</label>
        </div>
        <div class="form-floating">
            <textarea class="form-control" placeholder="Leave a comment here" id="comment"
                style="height: 100px"></textarea>
            <label for="floatingTextarea2">응원댓글</label>
        </div>
        <button onclick="save_comment()" type="button" class="btn btn-dark">댓글 남기기</button>
    </div>
    <div class="mycards" id="comment-list">
        <div class="card">
            <div class="card-body">
                <blockquote class="blockquote mb-0">
                    <p>새로운 앨범 너무 멋져요!</p>
                    <footer class="blockquote-footer">호빵맨</footer>
                </blockquote>
            </div>
        </div>
        <div class="card">
            <div class="card-body">
                <blockquote class="blockquote mb-0">
                    <p>새로운 앨범 너무 멋져요!</p>
                    <footer class="blockquote-footer">호빵맨</footer>
                </blockquote>
            </div>
        </div>
    </div>

```



```
        </div>
      </div>
      <div class="card">
        <div class="card-body">
          <blockquote class="blockquote mb-0">
            <p>새로운 앨범 너무 멋져요!</p>
            <footer class="blockquote-footer">호행맨</footer>
          </blockquote>
        </div>
      </div>
    </div>
  </body>
</html>
```

[이전 주차](#)

[다음 주차](#)

---

Copyright © TeamSparta All rights reserved.