



# BangEZ.com

## 오피스텔 파트

## 프로토타입 소개

- BangEZ 팀 개발자 노태호

## 개발 스택 및 주요 기능

개발 스택	핵심 기능	추가 구현 기능	
<p>1. 프론트 : Next.js 14, React 17</p> <p>1) 디자인 : Material-UI를 적극 활용</p> <p>2) 서버와의 통신 :</p> <ul style="list-style-type: none"><li>- 비동기 통신을 기본으로, 특정 부분은 동기로 처리.</li><li>- Axios와 Redux 활용</li></ul> <p>2. 서버 : Spring Boot 3</p> <ul style="list-style-type: none"><li>- JPQL, QueryDSL을 복합적으로 사용</li><li>- 보안이나 추가 기능이 필요없는 부분은 모두 QueryDSL로 속도에 주안점을 맞춰서 개발</li></ul>	<p>이번 프로토 타입의 핵심 기능은 QueryDSL의 동적쿼리 기능을 이용한 실시간 필터링 기능입니다.</p> <p>체크박스를 이용하여 매매, 전세, 월세 등의 조건을 실시간으로 변경할 수 있으며, 가격 범위 검색 기능도 함께 제공하고 있습니다.</p>	<p>〈유저〉</p> <ul style="list-style-type: none"><li>- 회원 가입</li><li>- 로그인</li><li>- JWT 발급</li><li>- 아이디/비밀번호 형식 실시간 체크 기능</li><li>- 회원 정보 수정</li><li>- 로그아웃/회원탈퇴</li></ul>	<p>〈오피스텔〉</p> <ul style="list-style-type: none"><li>- 실시간 매물 로딩</li><li>- 매물 등록<ul style="list-style-type: none"><li>+ 최상단으로 올리기</li></ul></li><li>- 카카오 API를 이용한 주소 검색</li><li>- 내가 올린 매물 보기<ul style="list-style-type: none"><li>+ 등록삭제</li></ul></li></ul>



BangEZ 로그인을 해주세요

ID : 대문자로 시작하는 6 ~ 13 문자열 (숫자, 기호 불가)

Little

영역에 맞는 ID

Password : 소문자로 시작 / 특수문자 하나 이상 포함 / 3 ~ 12 문자열

\*\*\*\*\*

사용할 수 있는 Password

로그인

BANGEZ 회원가입

## 회원 관리 기능



### 회원 가입

사용자는 간단한 절차를 통해 회원 가입을 할 수 있습니다.  
실시간 유효성 검사를 통해 안전한 회원 가입을 보장합니다.



### 로그인

회원 가입 후 로그인하면 마이페이지에 접근할 수 있습니다.  
로그인 정보는 안전하게 관리되며 로그아웃, 회원탈퇴 기능도 제공합니다.



### 마이페이지

마이페이지에서는 회원 정보 수정 기능을 이용할 수 있습니다.  
(PatchMapping을 이용하여 안전하게 처리)  
회원 탈퇴 기능도 제공하여 개인정보 보호를 강화하였습니다.



### 보안 및 데이터 관리

회원 정보와 매물 데이터는 데이터베이스에 안전하게 관리되며, 로그인 및 회원 탈퇴 기능을 통해 사용자의 개인정보를 보호하고 있습니다.

# 매물 등록 및 관리

## 매물 등록

사용자는 직접 매물 정보를 입력하여 등록할 수 있습니다. 카카오 API를 이용한 주소 검색 기능을 통해 편리하게 매물 위치를 설정할 수 있으며, 가격, 면적 등의 정보를 입력할 수 있습니다.

주소: 서울특별시 강남구

면적: 1000000

가격: 1000000

등록: 2024-05-03

삭제: 2024-05-03

주소: 서울특별시 강남구

면적: 1000000

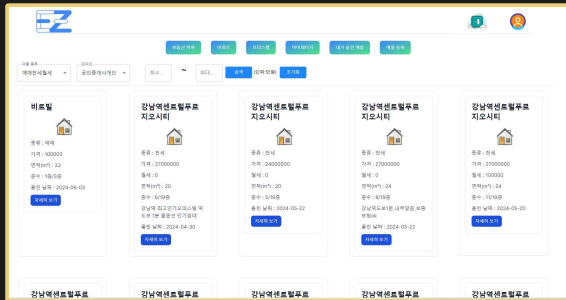
가격: 1000000

등록: 2024-05-03

삭제: 2024-05-03

비트빌 매물을 등록하고 '내가 올린 매물' 메뉴를 클릭한 사진

# 매물 등록 및 관리



## 매물 관리

등록된 매물은 '내가 올린 매물' 메뉴에서 확인할 수 있으며, 필요에 따라 수정 및 삭제가 가능합니다. 최근 등록한 매물이 상단에 표시되어 사용자의 편의성을 높였습니다.

## 추가 기능 계획

향후 매물 사진 등록, 매물 수정, 즐겨찾기, 알림 기능 등 다양한 기능을 추가할 계획입니다. 사용자의 편의성과 만족도를 높이기 위해 지속적으로 개선해 나갈 것입니다.



# 실시간 필터링 기능

1

## 체크박스 활용

매매, 전세, 월세 등의 조건을 체크박스로 제공하여 사용자가 실시간으로 선택할 수 있습니다.

3

## 동적 쿼리 DSL

QueryDSL을 활용하여 효율적이고 유연한 실시간 필터링 기능을 구현하였습니다.

The screenshot displays a real estate search interface. At the top, there are dropdown menus for '거래 종류' (Transaction Type) set to '월세' (Monthly Rent) and '지역' (Area) set to '개원' (Open). To the right are input fields for '최소' (Minimum) and '최대' (Maximum) price, followed by a '검색' (Search) button and a '(단위:만원)' (Unit: 10,000 Won) label. Below the search bar, three property listings are shown:

- 서초현대타워**
  - 종류: 월세
  - 가격: 80000000
  - 월세: 85000
  - 면적(m²): 41
  - 층수: 10/30층
  - 유효 날짜: 2024-05-24
  - 자세히 보기
- 강남역센트럴푸르지 오시티**
  - 종류: 월세
  - 가격: 27000000
  - 월세: 15000
  - 면적(m²): 24
  - 층수: 8/19층
  - 유효 날짜: 2024-05-18
  - 자세히 보기
- 강남역센트럴푸르지 오시티**
  - 종류: 월세
  - 가격: 27000000
  - 월세: 15000
  - 면적(m²): 24
  - 층수: 4/19층
  - 유효 날짜: 2024-05-23
  - 자세히 보기

2

## 가격 범위 검색

가격 범위를 직접 입력할 수 있는 기능을 통해 사용자가 원하는 가격대의 매물만 표시됩니다.

# 실시간 필터링 기능(코드)

```
export const findOfficetelsBoundaryAPI = async (
  searchInfo: OfficetelSearch
) => {
  const queryString = qs.stringify({
    q: "2",
    oValue: searchInfo.OValues,
    pValue: searchInfo.PValues,
    cost: searchInfo.cost,
  }, { arrayFormat: 'repeat' });

  const url = `/officetel/search?${queryString}`;
  try {
    const response = await instance().get(url);

    return response.data.list;
  } catch (error) {
    return error;
  }
};
```

```
@GetMapping(path = @"/search") & notValid = true
public ResponseEntity<> searchPlayer(
  @RequestParam(value = "q", required = true) String q,
  @RequestParam(value = "dto", required = false) OfficetelDTO dto,
  @RequestParam(value = "id", required = false) Long id,
  @RequestParam(value = "oValue", required = false) List<String> oValue,
  @RequestParam(value = "pValue", required = false) List<String> pValue,
  @RequestParam(value = "cost", required = false) List<String> cost,
  @RequestParam(value = "user", required = false) String user,
  Pageable pageable
) {
  log.info("id" + id);
  long totalCount = officetelService.countOfficetel();
  PageDTO pageDto = pageService.getPageDTO(totalCount, pageable.getPageSize(), pageable.getPageNumber());
  List<?> list = router.execute(q, dto, oValue, pValue, cost, pageable, id, user);

  Box box = new Box();
  box.setPageDTO(pageDto);
  box.setList(list);

  return ResponseEntity.ok(box);
}
```

```
public List<OfficetelDTO> getOfficetelDynamic(OfficetelDTO dto, List<String> oValue, List<String> pValue, List<String> cost) {
  BooleanBuilder oValueBuilder = new BooleanBuilder();
  BooleanBuilder pValueBuilder = new BooleanBuilder();
  BooleanBuilder costBuilder = new BooleanBuilder();

  if (!oValue.isEmpty()) {
    for (String i : oValue) {
      oValueBuilder.or(officetel.ownerType.eq(Long.valueOf(i)));
    }
  }

  if (!pValue.isEmpty()) {
    for (String i : pValue) {
      pValueBuilder.or(officetel.propertyType.eq(Long.valueOf(i)));
    }
  }

  if (!cost.isEmpty()) {
    long lowCost = Long.parseLong(cost.get(0))*10000;
    long maxCost = Long.parseLong(cost.get(1))*10000;
    costBuilder.and(officetel.price.gt(lowCost));
    costBuilder.and(officetel.price.lt(maxCost));
  }

  BooleanBuilder finalBuilder = oValueBuilder.and(pValueBuilder).and(costBuilder);

  return factory.selectFrom(officetel).RAQuery<OfficetelModel>
    .where(finalBuilder)
    .orderBy(officetel.id.desc())
    .fetch() List<OfficetelModel>
    .stream() Stream<OfficetelModel>
    .map(OfficetelDAOImpl::entityToDto) Stream<OfficetelDTO>
    .toList();
}
```

```
public List<?> execute(String type, OfficetelDTO dto, List<String> oValue, List<String> pValue, List<String> cost, Pageable pageable, Long id, String user) {
  return switch (type){
    case "1" -> service.getAllOfficetel(pageable);
    case "2" -> service.getOfficetelDynamic(dto, oValue, pValue, cost);
    case "3" -> service.getAllOfficetelNoPage();
    case "4" -> service.getOfficetelById(id);
    case "5" -> service.getOfficetelByUser(user);
    default -> null;
  };
}
```

# 감사합니다.

유튜브 시연 영상 : <https://youtu.be/nTSPLGzGMIQ?si=NZZ-667SwmIPs8lu>

소스 코드 :

frontend : [https://github.com/NohTaeHo1/bitcamp\\_officetel\\_front.git](https://github.com/NohTaeHo1/bitcamp_officetel_front.git)

backend : [https://github.com/NohTaeHo1/bitcamp\\_officetel\\_CRUD.git](https://github.com/NohTaeHo1/bitcamp_officetel_CRUD.git)