# <알고리즘 적용 기획서 #1>

PJT명: Happy House Project

단계: 알고리즘 적용 PJT

진행 일자: 2022.05.06

소속: SSAFY 7기 대전 7반

이름: 김구연, 송선아, 전승준, 조다연

### 1. 내용

-가격 순(오름차순, 내림차순)으로 정렬, 아파트 명 혹은 법정동으로 검색 하여 데이터를 보여지고 위도, 경도를 이용하여 아파트의 위치를 표시해주는 서 비스

## 2. 적용 알고리즘

- 가격 순(오름차순, 내림차순)으로 정렬하여 사용자에게 보여줌
- -아파트 명과 법정동을 검색하여 해당 검색어만 필터링하여 나타냄
- -지도에 매핑을 한 위치의 반경 500m내의 아파트 정보를 표시함

### 3. 알고리즘 개요

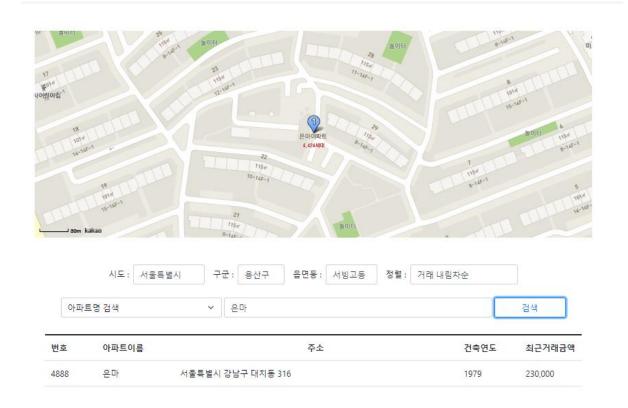
- 사용자가 본인의 기호에 따라 높은 거래 금액 혹은 낮은 거래 금액을 한눈에 쉽게 알아볼 수 있는 편의성 제공을 위해 아파트 매매값을 나열하여 보여지게된다.



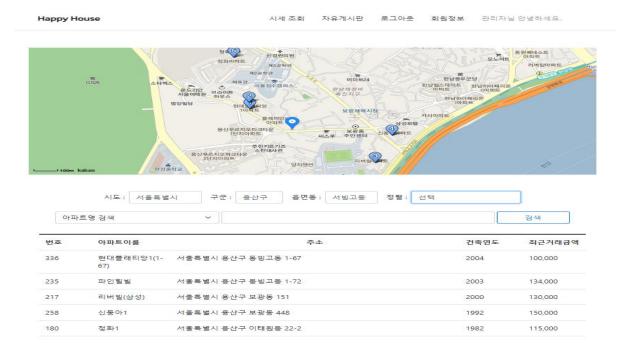


- 원하는 아파트 명과 법정동을 검색하여 해당 단어만 필터링 되어 데이 터를 나타내 사용자가 원하는 아파트의 정보를 알수 있다.





- 사용자가 원하는 지역에 어떠한 아파트가 위치해 있는지 보여질 수 있 도록 위치를 마우스로 매핑시키면 해당 위치의 반경 500내의 아파트의 정보들을 보여줍니다.



#### 4. 적용 서비스

- 아파트 명, 법정동 검색과 가격 순(내림차순, 오름차순) 정렬 매핑을 통해 주변 아파트 알림 서비스를 사용자에게 제공합니다.

## 5. 적용 서비스 개발 개요

- 아파트 명, 법정동 검색과 가격 순대로 나열하도록 개발하였습니다.

```
@GetMapping("/sortsearch")
public ResponseEntity<List<HouseInfoDto>> selectsort(
        @RequestParam("key") String key, @RequestParam("word") String word,
         @RequestParam("dong") String dong, @RequestParam("sort") String sort) throws Exception {
    List<HouseInfoDto> list = null;
    // word먼저. word가 null이 아니라면
    if(!word.equals("")) {
   // 그리고 aptName으로 아파트명으로 검색을 했다면
         if(key.equals("aptName")) {
             System.out.println("아파트명!" + word);
             list = haHouseMapService.viewHouseByName(word);
         // 법정동으로 검색을 했다면
         }else {
             System.out.println("법정동명! " + word);
             list = haHouseMapService.viewHouseByDong(word);
        }
    // word가 널이라면 -> dong 검색
    }else {
        System.out.println("등명! " + dong);
        list = haHouseMapService.getAptInDong(dong);
    // 오름차순
    if(sort.equals("dealasc")) {
        Collections.sort(list, new Comparator<HouseInfoDto>() {
             @Override
             public int compare(HouseInfoDto o1, HouseInfoDto o2) {
                 String price2 = o2.getRecentPrice().replace(" ", "").replace(",", "");
String price1 = o1.getRecentPrice().replace(" ", "").replace(",", "");
                 return Integer.parseInt(price1) - Integer.parseInt(price2);
        });
    }else { // 내림차순
        Collections.sort(list, new Comparator<HouseInfoDto>() {
             String price1 = o2.getRecentPrice().replace(" ", "").replace(",", "");
String price2 = o1.getRecentPrice().replace(" ", "").replace(",", "");
                 return Integer.parseInt(price1) - Integer.parseInt(price2);
        });
    return new ResponseEntity<List<HouseInfoDto>>(list, HttpStatus.OK);
}
```

```
<!-- 아파트명으로 검색하기 -->
    <select id="viewHouseByName" parameterType="String"</pre>
       resultType="HouseInfoDTO">
        select * from hahaha where aptName LIKE CONCAT('%', #{houseName}, '%') and recentPrice is not null
    <select id="getAllHouse" resultType="HouseInfoDto">
        select h.aptCode, h.aptName, h.buildyear, h.dongCode, h.dongName, h.jibun, h.lat, h.lng, si.sidoname, gu.gugunname,
                (select dealAmount from housedeal where aptCode = h.aptCode and no = (select max(no) from housedeal where aptCode = h.aptCode)) recentPrice
        from houseinfo h
        left join sidocode si
       on left(h.dongcode,2) = left(si.sidocode,2)
       left join guguncode gu
       on left(h.dongcode,5) = left(gu.guguncode,5)
    </select>
    <!-- 동명으로 검색하기 -->
    <select id="viewHouseByDong" parameterType="String"</pre>
        resultType="HouseInfoDTO">
        select * from hahaha where dongName LIKE CONCAT('%', #{dongName}, '%') and recentPrice is not null
        <!-- select h.aptCode, h.aptName, h.buildyear, h.dongCode, h.dongName,
        h.jibun, h.lat, h.lng, si.sidoname, gu.gugunname,
        (select dealAmount from housedeal where aptCode = h.aptCode and no = (select
        max(no) from housedeal where aptCode = h.aptCode)) recentPrice
        from houseinfo h
        left join <u>sidocode</u> si
        on left(h.dongcode,2) = left(si.sidocode,2)
        left join guguncode gu
       on left(h.dongcode,5) = left(gu.guguncode,5) where h.dongName LIKE CONCAT('%', #{dongName}, '%')
       order by aptName; -->
    </select>
</mapper>
```

- 6 -