

※ 프로젝트명 클릭시 해당 플랫폼 이동

스마트 그린도시 플랫폼

동구라미



참여기간

- 2023.10 ~ 진행중

개발 환경

- Web Backend(Java, SpringBoot, MyBatis, MariaDB)
- Web frontend(React, TypeScript)
- Versioning(Gitlab)
- DevOps(회사내의 폐쇄망 서버 (개발) 배포 및 테스트 진행)

협업 환경

- 개발 프로세스: google spreadsheet 를 이용하여 요구기능 및 진행 이력 관리
- - 문서화 : Excel문서 활용

담당업무

- Web backend
 - Rest 방식의 API 개발
 - DB 설계 및 커스텀
 - 개발서버 세팅 및 프로젝트, Postman 배포

- 일반메뉴
- 권한필요메뉴
- 웹 전용메뉴





로그인 | 회원가입

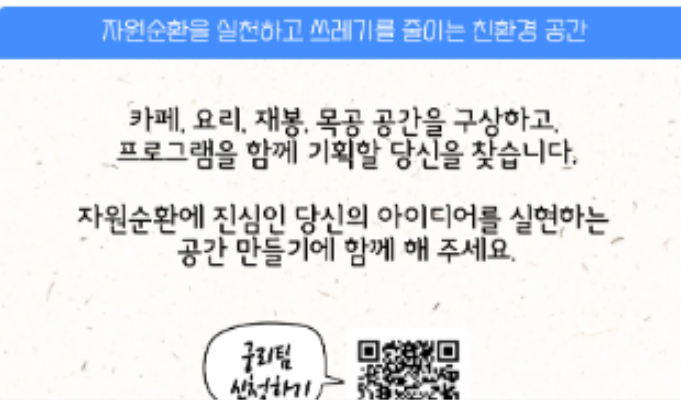
클린동구 친환경 자원순환센터 그린도시지도 그린도시이야기 그린도시포인트

센터 소식

검색어를 입력해주세요

[모집] 친환경자원순환센터 공리팀을 모집합...

2024년 개관하는 친환경자원순환센터에서는 무엇을 하면 좋을까요? 각 공간마다 어떤 특색을 담아 프로...



자원순환 마을이 되면?? 🤔 🤔

[스마트 그린] 꼬두메 자원순환 시범마을 자원순환 마을이 되면 이렇게 바뀔니다! 1. 재활용품, 종량제 쓰레...



스마트그린도시 꼬두메 자원순환 시범마을 ...

여러분은 스마트그린도시사업에 대해서 아시나요? 스마트 그린도시는 기후·환경 위기 속에서 지속가능한 ...



게시판

Spring Boot 의 MyBatis 프레임워크를 활용하여 직접적인 쿼리 튜닝을 통해 게시물에 대한 기본 CRUD, 이미지 및 파일 다운,업로드 정렬 및 필터링, 댓글, 좋아요 등 게시판 기능을 구현




지도기반 게시판

좌표를 활용하여 해당위치
나의 행정동 환경정화를
위한 위치신고 서비스
Spring Boot 의 MyBatis
프레임워크를 활용하여 직접
적인 쿼리 튜닝을 통해지도
기반 게시물에 대한기본
CRUD, 이미지 업로드
정렬 및 필터링 기능을 구현



관리자 페이지

투닝된 쿼리를 통해 조회한
위치 신고 게시물의 통계
결과를 관리자 권한을
부여 받은 유저의 화면에서
실시간 현황을 필터링하여
조회할 수 있도록 구현한
관리자 계정 대시보드 화면



로그아웃 마이페이지 1,020 P

클린동구 친환경 자원순환센터 그린도시지도 그린도시이야기 그린도시포인트

포인트 조회 및 교환

ⓘ 1,000 P 부터 교환 가능

보유 포인트 1,020 P

포인트 교환신청

1주일 1개월 3개월 2023-10-08 ~ 2024-01-08 조회

일시	분야	내용	구분	점수	소멸 예정일시
2024.01.05	회수	회수	회수포인트	-1,000 P	2028.12.05
2023.12.08	스마트그린도시플랫폼	테스트	참여포인트	1,000 P	2028.12.08
2023.12.06	포인트 교환	1,000 포인트 교환신청	사용포인트	-1,000 P	-

포인트

쓰레기 위치신고 서비스,
분리수거 등을 통해 적립된
포인트 내역, 교환 서비스
Spring Boot 의 MyBatis
프레임워크를 활용하여
직접적인 쿼리 튜닝을 통해
지도 기반 게시물에 대한
기본 CRUD, 이미지 업로드
정렬 및 필터링 기능을 구현


```

@Autowired
CustomOAuth2UserService customOAuth2UserService;

private final LoginSuccessHandler loginSuccessHandler;

0개의 사용위치  👤 V2seon *
@Bean
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
    http.csrf().disable();

    http.authorizeRequests() ExpressionInterceptUrlRegistry
        .antMatchers( ...antPatterns: "/member/**").permitAll()
        .antMatchers( ...antPatterns: "/comap/**").permitAll()
        .anyRequest().authenticated()

        .and() HttpSecurity
        .oauth2Login() OAuth2LoginConfigurer<HttpSecurity>
        .loginPage("/")
        .successHandler(loginSuccessHandler)
        .userInfoEndpoint() UserInfoEndpointConfig
        .userService(customOAuth2UserService);

    http.logout()
        .logoutRequestMatcher(new AntPathRequestMatcher( pattern: "/logout"))
        .logoutSuccessUrl("/main")
        .invalidateHttpSession(true);

    return http.build();
}

```

Spring Security

Security를 통해 소셜 로그인을 구현하고 로그인시 Session 을 생성하여 허가되지 않은 상태에서는 API 접근을 할 수 없도록 예외처리 및 별도의 회원가입 로직 구현

```

/**
 * 일반_등록_성공
 */
public final static String COMMON_INSERT_OK= "정상적으로 등록되었습니다.";
/**
 * 일반_수정_성공
 */
public final static String COMMON_UPDATE_OK= "정상적으로 수정되었습니다.";
/**
 * 일반_삭제_성공
 */
public final static String COMMON_DELETE_OK= "정상적으로 삭제되었습니다.";
/**
 * 일반_탐색_성공
 */
public final static String COMMON_SEARCH_OK= "정상적으로 탐색되었습니다.";
/**
 * 일반_전송_성공
 */
0개의 사용위치
public final static String COMMON_SEND_OK= "정상적으로 전송되었습니다.";
/**
 * 일반_전송_실패
 */
0개의 사용위치
public final static String COMMON_SEND_FAIL= "전송에 실패하였습니다.";
/**
 * V2seon, 2023-08-29 오후 4:32 • 최초
 * 0개의 사용위치
 */

```

Retrun Message 모듈

API의 기능에 따라서 실행하는 기능의 리턴하는 결과 값 또는 예외처리 내용을 분류해 놓은 모듈

```
SELECT
NM.MBR_UID,
TN.TITLE,
CASE NMA.AUTH_ROLE
WHEN 'AMN' THEN '총괄관리자'
WHEN 'MST' THEN '공무원'
WHEN 'SMT' THEN '센터관리자'
WHEN 'ABR' THEN '센터직원'
WHEN 'MBR' THEN '쓰레기해결단'
WHEN 'STB' THEN '주민'
ELSE '알 수 없음'
END AS 'auth',
NM.nickName,
(SELECT COUNT(*) FROM CM_MAPPER WHERE CRT_UID = NM.MBR_UID
AND CM_UID IN (SELECT CM_UID FROM CM_MAPPING_CATEGORY WHERE CMC_UID = #{cmcUid})
AND CM_STATUS = 'A' AND DATE(CRT_DT) >= #{startDt} AND #{endDt} >= DATE(CRT_DT)) AS 'ATC',
(SELECT COUNT(*) FROM CM_MAPPER WHERE CRT_UID = NM.MBR_UID
AND CM_UID IN (SELECT CM_UID FROM CM_MAPPING_CATEGORY WHERE CMC_UID = #{cmcUid})
AND CM_STATUS = 'B' AND DATE(CRT_DT) >= #{startDt} AND #{endDt} >= DATE(CRT_DT)) AS 'BTC',
(SELECT COUNT(*) FROM CM_MAPPER WHERE CRT_UID = NM.MBR_UID
AND CM_UID IN (SELECT CM_UID FROM CM_MAPPING_CATEGORY WHERE CMC_UID = #{cmcUid})
AND DATE(CRT_DT) >= #{startDt} AND #{endDt} >= DATE(CRT_DT)) AS 'TTC'
FROM NU_MEMBER_AUTH NMA
INNER JOIN TN_COMMUNITY TN ON TN.TC_UID = NMA.TC_UID
INNER JOIN NU_MEMBER NM ON NMA.MBR_UID = NM.MBR_UID
WHERE NMA.AUTH_ROLE != 'NON'
<if test="auth != null">
    AND NMA.AUTH_ROLE = #{auth}
</if>
<if test="tcUid != null">
    AND TN.TC_UID = #{tcUid}
</if>
ORDER BY TTC DESC
```

유저들의 플랫폼 활동 통계를 위한 쿼리

'< if >' 태그를 사용하여 조건이 동적으로 변경되게 하였으며
서브쿼리를 사용해 사용자에게 대한 특정 조건에 따른 값을 가져오고
매개변수를 활용해 동적으로 값이 전달되도록 구현

```
public int markCoordinate(BigDecimal latitude, BigDecimal longitude) throws Exception {
    int result = 0;

    // 카카오 Geocoding API 요청 URL (API 키를 포함해야 합니다)
    String apiKey = "006af5dd14cd729e7a4b3a9a2861cdca"; // 본인의 API 키로 교체
    String apiUrl = "https://dapi.kakao.com/v2/local/geo/coord2regioncode.json?x=" + longitude + "&y=" + latitude;

    try {
        URL url = new URL(apiUrl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Authorization", "KakaoAK " + apiKey); // API 키 추가

        // 응답 읽기
        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        StringBuilder response = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            response.append(line);
        }
        reader.close();

        // JSON 파싱
        String jsonResponse = response.toString();
        JSONObject jsonObject = JsonParser.parseString(jsonResponse).getAsJsonObject();

        // 원하는 데이터 추출
        String regionName = jsonObject.getAsJsonArray("documents")
            .get(1).getAsJsonObject()
            .get("region_3depth_name").getString();

        // 주소 출력
        TnCommunityVO tnCommunityVO = new TnCommunityVO();
        tnCommunityVO.setTitle(regionName);
        TnCommunityVO getdata = tnCommunityService.getTcTitle(tnCommunityVO);
        if (getdata != null) result = getdata.getTcUId();
        if (getdata == null) throw new IllegalArgumentException("해당 위치는 등록 불가합니다.");
    }
}
```

허가된 행정동인지 확인을 위한 지오코딩 모듈

위치 신고시 위도, 경도값을 받아와 해당 위치의 해당되는
행정동명을 찾아내어 DB에 등록된 행정동 목록에 있는 행정동이
아닐경우 예외 처리를 하는 모듈

Thank you for enjoying

Development Portfolio

by Lee Seon Jae

PHONE 010.2221.0325

Github <https://github.com/V2seon>

E-Mail lsj22210325@gmail.com