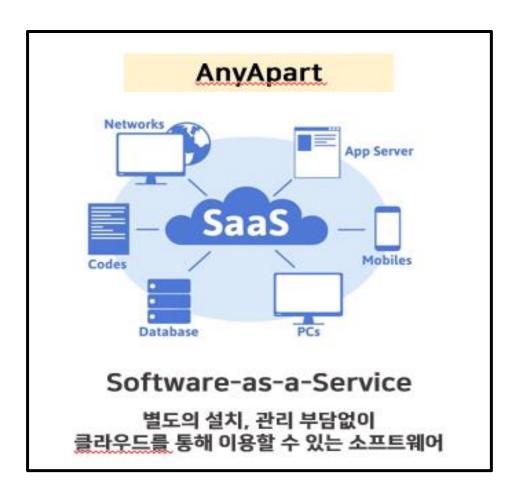
# 아파트통합관리시스템 AnyApart

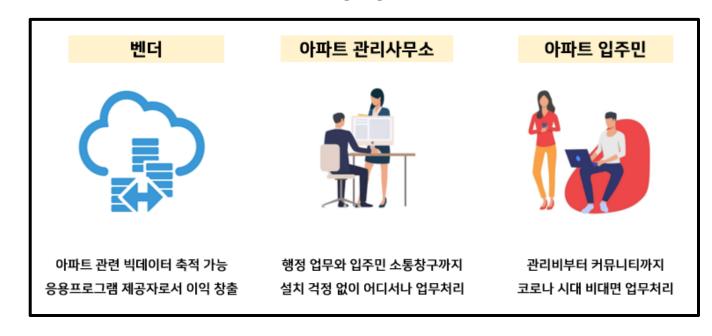
### 주제선정배경

우리 생활과 밀접한 프로그램을 만들고자하던 중, 우리나라 인구의 절반이 아파트에 거주하고 있다는 기사를 접하게 되었습니다. 기존 아파트 관리시스템을 이용하기 위해서는 별도의 프로그램을 설치해야 했기 때문에 사용자로서는 다소 번거로운 부분이 있었습니다.

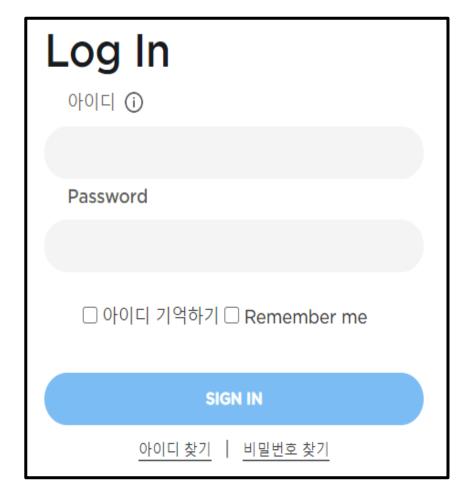
이러한 개선점을 보완하기 위하여 우리 프로젝트에서는 SaaS(Software-as-a-Service)를 지향, 별도의 설치나 관리의 부담 없이 클라우드를 통해 이용할 수 있는 아파트 관리시스템을 만들고자 하였습니다.



### 사이트



# 로그인 – 로그인, 비밀번호 변경



비밀번호 변경



```
<beans:bean id="authenticationProvider"
    class="org.springframework.security.authentication.dao.DaoAuthenticationProvider"
    p:userDetailsService-ref="customUserService"
    p:passwordEncoder-ref="passwordEncoder"
    p:hideUserNotFoundExceptions="false" />
```

```
<form-login login-page="/login"
    username-parameter="memId" password-parameter="memPass"
    login-processing-url="/login/loginProcess.do"
    authentication-success-handler-ref="customAuthenticationSuccessHandler"
    authentication-failure-url="/login" />
```

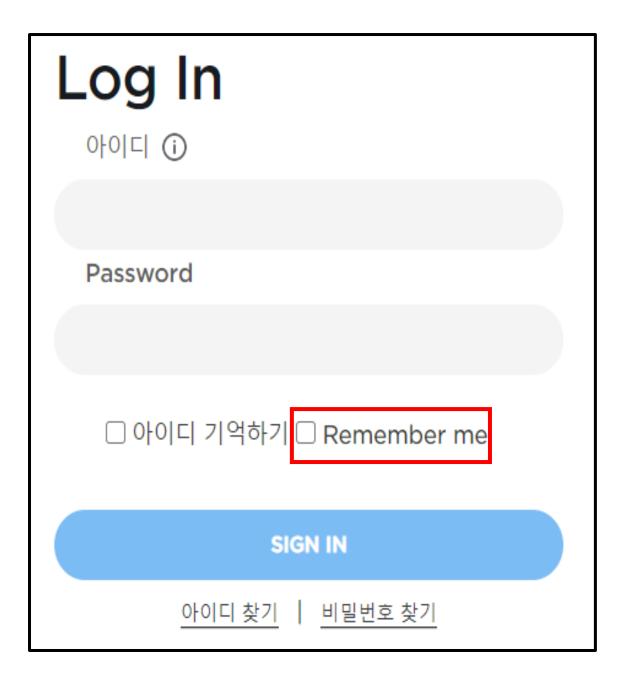
### 회원 권한 메인 페이지 이동

### Login-success-handler

```
public class CustomAuthenticationSuccessHandler extends SavedRequestAwareAuthenticationSuccessHandler{
  @Inject
   private IAuthDAO dao;
   protected void handle(HttpServletRequest request, HttpServletResponse response, Authentication authentication)
           throws IOException, ServletException {
       String mem_id = authentication.getName();
       MemberVO memberVO = dao.authMember(mem_id);
       WebUtils.setSessionAttribute(request, "memId", memberV0.getMemId());
       String tmp = (String) WebUtils.getSessionAttribute(request, "memId");
       if(authentication.getAuthorities().contains(new SimpleGrantedAuthority("ROLE_RES"))) {
           setDefaultTargetUrl("/resident");
       }else if(authentication.getAuthorities().contains(new SimpleGrantedAuthority("ROLE_ADMIN"))) {
          setDefaultTargetUrl("/vendor");
       }else {
          setDefaultTargetUrl("/office");
       super.handle(request, response, authentication);
```

## 로그인 – Remember me

### 로그인 정보 기억하기



```
<remember-me remember-me-parameter="rememberMe"
    remember-me-cookie="rememberMeCookie" data-source-ref="dataSource"
    user-service-ref="customUserService" />

<intercept-url pattern="/login/**" access="permitAll" />

<intercept-url pattern="/office/setting/**" access="hasRole('ROLE_HEAD')"/>

<intercept-url pattern="/vendor/**" access="hasRole('ROLE_ADMIN')"/>

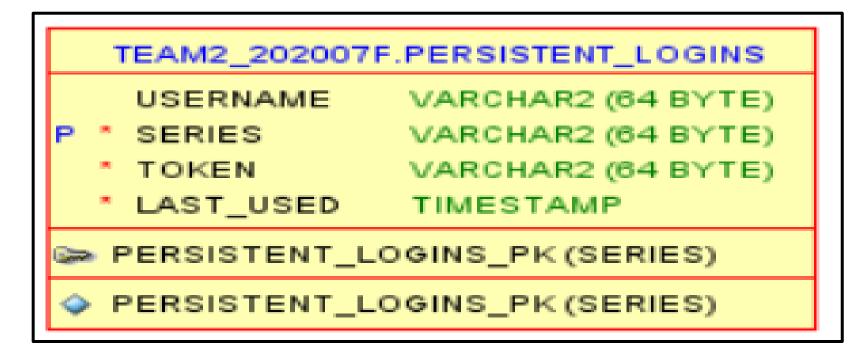
<intercept-url pattern="/resident/**" access="hasRole('ROLE_RES')"/>

<intercept-url pattern="/**" access="permitAll" />

<intercept-url pattern="/**" access="permitAll" />

<access-denied-handler error-page="/AccessDenied.do" /></a>
```

#### DB 모델



PERSISTAENT\_LOGINS 테이블의 정보를 가지고 인증 토큰을 생성해줍니다.

# 로그인 – ID, PW 찾기



#### 메일 API와 구글 SMTP 서버이용

```
public String mailSend(MemberVO member) {
   ServerSpec mailServer = ServerKind.GMAIL.getSmtp();
   String username = "hkjs9620";
   String password = "gqhqpumkxhgfdfke";
   String to = member.getMemEmail();
   String from = "AnyApart";
   // 1) MailSender 기본설정
   JavaMailSenderImpl mailSender = new JavaMailSenderImpl();
   mailSender.setHost(mailServer.getHost());
   mailSender.setPort(mailServer.getPort());
   mailSender.setUsername(username);
   mailSender.setPassword(password);
   Properties props = new Properties();
   props.put("mail.smtp.starttls.enable", true);
    mailSender.setJavaMailProperties(props);
   String certNum = RandomStringUtils.randomNumeric(6);
   // 2) SimpleMailMessage를 이용한 메시지 설정
   SimpleMailMessage mail = new SimpleMailMessage();
   mail.setTo(to);
   mail.setFrom(from);
   mail.setSubject("AnyApart ID/PW 찾기 인증번호");
   mail.setText("인증번호: " + certNum);
   // 3) 전송
   mailSender.send(mail);
   System.out.println("전송 완료");
   return certNum;
```

# 민감정보 암호화 – Jasypt 이용

### Pom.xml dependency

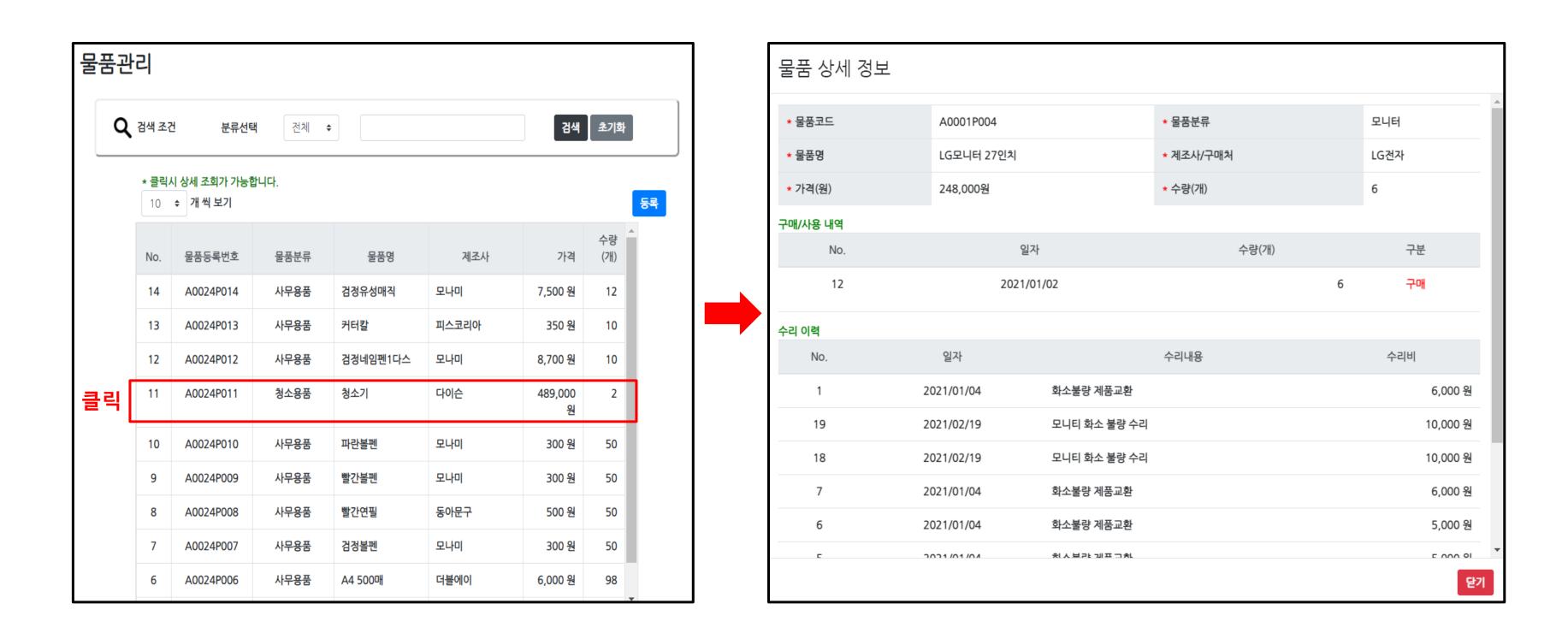
#### Bean 설정

```
<bean id="encryptorConfig" class="org.jasypt.encryption.pbe.config.EnvironmentPBEConfig">
    kproperty name="algorithm" value="PBEWithMD5AndDES" />
    kproperty name="password" value="java" />
</bean>
<bean id="encryptor" class="org.jasypt.encryption.pbe.StandardPBEStringEncryptor">
    cproperty name="config" ref="encryptorConfig" />
</bean>
<bean class="org.jasypt.spring31.properties.EncryptablePropertyPlaceholderConfigurer">
    <constructor-arg ref="encryptor" />
        cproperty name="locations" value="classpath:kr/or/anyapart/db/dbInfo.properties">
   </property>
</bean>
<bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource"</pre>
   p:driverClassName="${maindb.driverClassName}"
   p:url="${maindb.url}"
   p:username="${maindb.user}"
   p:password="${maindb.password}"
   p:initialSize="${maindb.initialSize}"
   p:maxTotal="${maindb.maxActive}"
    p:maxWaitMillis="${maindb.maxWait}"
```

#### dbInfo.properties

```
7 maindb.url=ENC(VXjGUQ70B+pWA7AMmLIAXggp/nJDp9CfEAJMqe9V6RyaoQxwSpJeT63tvciUPrl8fU8zKLBsFkyNByTj6qtdKQ==)
8 maindb.user=ENC(o432phE0oMiZkAItzKuK0lJm3nb4cex2)
9 maindb.password=ENC(8kY+ajL9VAI8rqIutdXeIw==)
```

# 물품 관리 – 물품 조회



클릭 시 해당 물품 정보의 상세보기를 모달 창으로 보여줍니다.

# 물품 관리 – 물품 등록

#### 여러 개 입력

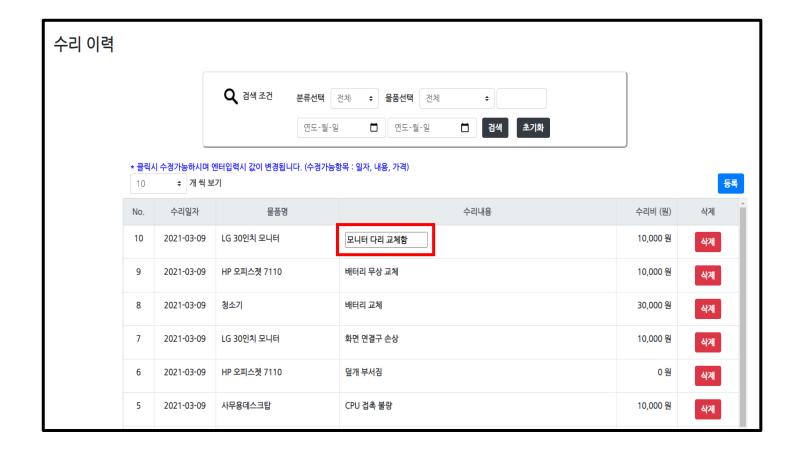


#### 추가/제거 하기

```
$("#plusBtn").on("click", function(){
   let clickir = $( #insertArea ).find( tr ).eq(0);
    let clickTr = $('#insertArea').find('tr').last();
    let newTr = clickTr.clone();
    let childInput = newTr.find(":input");
    let name;
    $(childInput).each(function(idx,element){
        // 각 input 값을 찾아서 name을 변경
        name = parseInt(element.name.substring(9,10));
        name = String(name + 1);
        str = element.name.replaceAt(9, name);
        element.setAttribute("name",str);
    let minus = newTr.find(".minusBtn");
    $(minus).show();
    let inputTag = newTr.find(":input");
    inputTag.val("");
    clickTr.after(newTr);
    newTr.find("input[type=hidden]").val("${pagingV0.searchV0.searchAptCode }");
});
$("#insertArea").on("click", ".minusBtn", function(){
    let deleteTr = $(this).parent().parent();
    deleteTr.prev().find(".plusBtn").show();
    deleteTr.detach();
});
```

# 물품 관리 – 수리 이력, 구매/사용내역

### 클릭 시 변경가능 상태









# Q&A 게시판

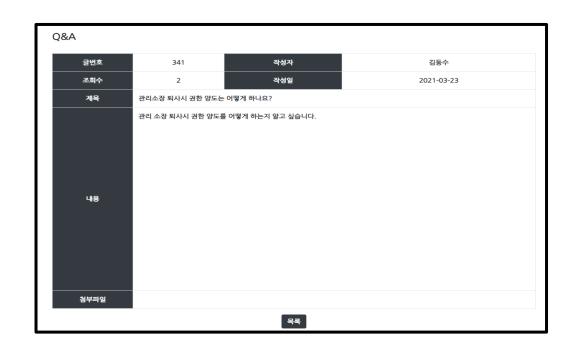
#### 벤더사이트 Q&A



#### 관리사무소 사이트 Q&A

				07
No.	세목	작성자	작성일	조회수
12	[일반]관리소장 퇴사시 권한 양도는 어떻게 하나요?	김동수	2021-03-23	1
11	[기능]결재 알림창에 추가 기능넣어주세요	이경륜	2021-03-10	33
10	[기능]급여 항목은 어떻게 보나요? <mark>답변</mark>	이경륜	2021-03-10	29
9	└ re : [기능]급여 항목	애니아파트	2021-03-10	28
8	[일반]검색은 어떻게 이용하나요?	이경륜	2021-03-10	40

### 관리사무소 사이트 게시글 상세 조회



#### 벤더 사이트 게시글 상세 조회



#### 계층형 게시글 구조

```
<select id="selectOfficeQnaList" parameterType="kr.or.anyapart.vo.PagingVO" resultType="kr.or.anyapart"</pre>
   <include refid="apartBoard"/>
   SELECT B.*
       FROM (
           SELECT A.*, ROWNUM RNUM
           FROM (
                  C.BO_NO, SUBSTR(CODE_NAME,5) AS BO_TYPE, C.BO_TITLE,
                  C.MEM_NICK AS BO_WRITER, C.BO_CONTENT, TO_CHAR(C.BO_DATE, 'YYYY-MM-DD')BO_DATE,
                  C.BO_HIT, C.BO_PARENT, C.BO_DEPTH
                  , E.BO_NO AS ANSWER_FLAG , C.APT_NAME
               FROM APARTBOARD C JOIN COMMONCODE D ON ( C.BO_TYPE = D.CODE_ID)
                              LEFT OUTER JOIN BOARD E ON (C.BO_NO = E.BO_PARENT)
               <include refid="searchFrag"/> <!-- 이거 동적으로 변경되야하는 부분 -->
               <include refid="unansweredFrag"/>
               START WITH C.BO PARENT IS NULL
               CONNECT BY PRIOR C.BO_NO = C.BO_PARENT 〈!-- 여기는 필수 -->
               ORDER SIBLINGS BY C.BO_NO DESC
       ) B
   <![CDATA[
       WHERE RNUM >= #{startRow} AND RNUM <= #{endRow}
</select>
```

# 시설 - 시설 예약



원하는 시간대를 드래그하면 모달 창으로 예약 정보가 넘어갑니다.

# 시설 - 시설 예약 프로시저

```
create or replace procedure p_reservation(
 p_resv_date IN varchar2, p_resv_start IN reservation.resv_start%type, p_resv_end IN reservation.resv_end%type,
 p_resv_cnt IN reservation.resv_cnt%type, p_cmnt_no IN COMMUNITY.CMNT_NO%type, p_mem_id IN member.mem_id%type, p_resv_no out number -- 성공하면 예약 번호를 내보냄,
 v_resv_no reservation.resv_no%type := 0; -- 예약 상세 변수
 v_apt_code apart.apt_code%type := NULL; -- 매개변수로 받은 멤버 값에서 아파트 코드를 추출해 넣어준다.
  capa number := 0;
 V_STARTFLAG NUMBER := 0; -- 30분에서 시작하는 지 정각에서 시작하는지 확인하는 플래그 변수
 V_ENDFLAG NUMBER := 0; -- 30분으로 끝나는지 정각에서 끝나는지 확인하는 플래그 변수
 V_TIME NUMBER := 0; -- 시각의 차이 * 2 를 담고 있을 변수
 V CNT NUMBER := 0; -- 반복하는 숫자를 담고 있을 변수
 V_CHANGE NUMBER := 0; -- 계산식을 변경할 녀석
 V_START_TIME VARCHAR2(4) := '00'; -- 시작하는 시간을 담고 있는다.
 V CHECK number := -1; -- 예약이 가능한지 불가능 한지 체크하는 녀석
begin
    (끝 시각 - 시작 시각) * 2 -1(30분에 시작시) +1(30분에 끝날시)
 SELECT CMNT_CAPA INTO capa FROM COMMUNITY WHERE CMNT_NO = P_cmnt_no; -- 최대 수용인원을 가져온다.
  -- 조회할 시작 시간과, 끝나는 시간을 확인해줘야한다. 변수를 만들어서 담아준다.
  -- 반복될 횟수도 계산해서 가지고 있어야한다.
  IF SUBSTR(p_resv_start,3,4) = '30' THEN
    V STARTFLAG := -1:
    -- 시작시간 30분이면 -1
    V_CHANGE := 1;
  ELSIF SUBSTR(p_resv_end,3,4) = '30' THEN
    V_ENDFLAG := -1;
     -- 끝나는 시간 30분이면 -1
 V_TIME := ( TO_NUMBER(SUBSTR(p_resv_end,1,2)) - TO_NUMBER(SUBSTR(p_resv_START,1,2)) ) * 2 ; -- 시각 차이 * 2 를 담음.
  -- 반복을 몇번이나 할 것인지 정함?
  IF V STARTFLAG < 0 THEN
    V_TIME := V_TIME - 1;
 ELSIF V_ENDFLAG < 0 THEN
   V_TIME := V_TIME + 1;
 V_START_TIME := substr(p_resv_start,1,2); -- 시작 시간을 담아준다.
  V_APT_CODE := substr(p_mem_id,1,5); -- 반복문 시작전에 아파트 코드 넣어주기
    EXIT WHEN V_TIME = V_CNT; -- 루프 다 돌았으면 나가기
    -- 30분 시작인지 정각 시작인지 정한다.
    IF V CHANGE = 1 THEN
      SELECT NVL(MAX(A.RESV CNT),0) AS RESV CNT
         into val
      FROM( SELECT
              B.CMNT NO
              , C.CMNT NAME
              , DETAIL START AS RESV START
              , DETAIL_END AS RESV_END
              , SUM(RESV_CNT) AS RESV_CNT
              RESVDETAIL A LEFT OUTER JOIN RESERVATION B ON (A.RESV_NO = B.RESV_NO)
                      LEFT OUTER JOIN COMMUNITY C ON (B.CMNT_NO = C.CMNT_NO)
           WHERE C.APT_CODE = v_apt_code
             and c.cmnt_no = p_cmnt_no
            GROUP BY B.CMNT_NO, C.CMNT_NAME, DETAIL_START, DETAIL_END
         ) A LEFT OUTER JOIN COMMUNITY B ON (A.CMNT_NO = B.CMNT_NO)
```

```
WHERE A.RESV CNT = B.CMNT CAPA
          and RESV_START = p_resv_date||'T'||V_START_TIME||':'||'30'
          and RESV_END = p_resv_date||'T'||LPAD(TO_NUMBER(V_START_TIME)+1,2,'0')||':'||'00';
                -- 30분 간격으로 인원을 조회한다.
       EXIT WHEN val + p_resv_cnt > capa ; --인원이 다 차있으면 예약 등록 취소
       V_{CHECK} := 1;
       SELECT NVL(MAX(A.RESV_CNT),0) AS RESV_CNT
       FROM( SELECT
               B.CMNT_NO
               , C.CMNT_NAME
               , DETAIL_START AS RESV_START
               , DETAIL_END AS RESV_END
               , SUM(RESV_CNT) RESV_CNT
               RESVDETAIL A LEFT OUTER JOIN RESERVATION B ON (A.RESV_NO = B.RESV_NO)
                        LEFT OUTER JOIN COMMUNITY C ON (B.CMNT_NO = C.CMNT_NO)
             WHERE C.APT_CODE = v_apt_code
              and c.cmnt_no = p_cmnt_no
             GROUP BY B.CMNT_NO, C.CMNT_NAME, DETAIL_START, DETAIL_END
         ) A LEFT OUTER JOIN COMMUNITY B ON (A.CMNT_NO = B.CMNT_NO)
       WHERE RESV_CNT = B.CMNT_CAPA
          and RESV_START = p_resv_date||'T'||V_START_TIME||':'||'00'
          and RESV_END = p_resv_date||'T'||V_START_TIME||':'||'30';
       -- 30분 간격으로 인원을 조회한다
       EXIT WHEN val + p_resv_cnt > capa ; --인원이 다 차있으면 예약 등록 취소
       V CHECK := 1;
       V_CHANGE := 1;
    END IF;
    V_CNT := V_CNT+1;
  end loop;
  p_resv_no := -1;
   - 실패시 들어갈 값
  IF V CHECK > 0 THEN
    SELECT MAX(RESV_NO)+1 AS RESV_NO INTO v_resv_no FROM RESERVATION;
    INSERT INTO reservation (
       resv_no,
       resv_date,
       resv_start,
       resv_end,
       cmnt no.
       mem_id
    ) VALUES (
       TO_DATE(P_RESV_DATE,'YYYY/MM/DD'),
       P RESV START,
       P RESV END,
       P_RESV_CNT,
       P_CMNT_NO,
       P_MEM_ID
    p_resv_no := V_RESV_NO;
      - 인서트 성공시 해당 예약번호 반환
  END IF;
end;
```

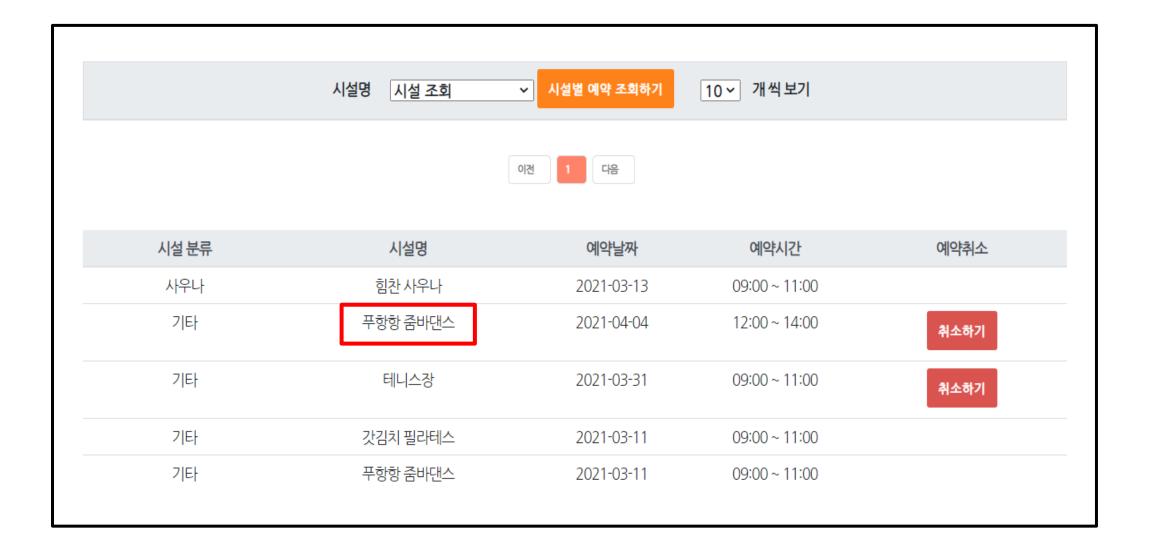
# 시설 - 시설 예약 세부내역 프로시저

```
create or replace procedure p_reservation_detail(
 p_resv_date in varchar2, -- 예약 날짜
 p_resv_start in varchar2, -- 예약 시작 시간
 p_resv_end in varchar2, -- 예약 끝나는 시간
 P_cmnt_no in number, -- 시설 번호
 p_resv_cnt in number, -- 예약인원
 p_resv_no in number, -- 예약 번호
  P_APT_CODE IN VARCHAR2 -- 아파트 코드
 ,p_result out number -- 성공했는지 못했는지 내보낼 결과값 양수면 예약이 성공한 것
 val number :=0;
 detailNo number := 0;
  V_STARTFLAG NUMBER := 0; -- 30분에서 시작하는 지 정각에서 시작하는지 확인하는 플래그 변수
  V ENDFLAG NUMBER := 0; -- 30분으로 끝나는지 정각에서 끝나는지 확인하는 플래그 변수
 V_TIME NUMBER := 0; -- 시각의 차이 * 2 를 담고 있을 변수
 V_CNT NUMBER := 0; -- 반복하는 숫자를 담고 있을 변수
 - V CHANGE NUMBER := 0; -- 계산식을 변경할 녀석
 V_START_TIME VARCHAR2(4) := '00'; -- 시작하는 시간을 담고 있는다.
begin
    (끝 시각 - 시작 시각) * 2 -1(30분에 시작시) +1(30분에 끝날시)
 SELECT CMNT_CAPA INTO capa FROM COMMUNITY WHERE CMNT_NO = P_cmnt_no; -- 최대 수용인원을 가져온다
  -- 조회할 시작 시간과, 끝나는 시간을 확인해줘야한다. 변수를 만들어서 담아준다.
  -- 반복될 횟수도 계산해서 가지고 있어야한다.
 IF SUBSTR(p resv start,3,4) = '30' THEN
   V_STARTFLAG := -1;
                             -- 30분이면 -1
    V CHANGE := 1;
  ELSIF SUBSTR(p_resv_end,3,4) = '30' THEN
   V_{ENDFLAG} := -1;
                           -- 30분이면 -1
  V_TIME := ( TO_NUMBER(SUBSTR(p_resv_end,1,2)) - TO_NUMBER(SUBSTR(p_resv_START,1,2)) ) * 2 ; -- 시각 차이 * 2 를 담음.
  -- 반복을 몇번이나 할 것인지 정함?
 IF V STARTFLAG < 0 THEN
   V TIME := V TIME - 1:
 ELSIF V_ENDFLAG < 0 THEN
   V_TIME := V_TIME + 1;
  V_START_TIME := substr(p_resv_start,1,2); -- 시작 시간을 담아준다.
   EXIT WHEN V_TIME = V_CNT; -- 루프 다 돌았으면 나가기
    -- 30분 시작인지 정각 시작인지 정한다.
    IF V CHANGE = 1 THEN
      SELECT NVL(MAX(A.RESV_CNT),0) AS RESV_CNT
         into val
              B.CMNT NO
              , C.CMNT NAME
              , DETAIL_START AS RESV_START
              , DETAIL END AS RESV END
               , SUM(RESV_CNT) AS RESV_CNT
              RESVDETAIL A LEFT OUTER JOIN RESERVATION B ON (A.RESV NO = B.RESV NO)
                     LEFT OUTER JOIN COMMUNITY C ON (B.CMNT_NO = C.CMNT_NO)
       WHERE C.APT CODE = P APT CODE
                    and c.cmnt_no = P_cmnt_no
                   GROUP BY B.CMNT_NO, C.CMNT_NAME, DETAIL_START, DETAIL_END
                ) A LEFT OUTER JOIN COMMUNITY B ON (A.CMNT_NO = B.CMNT_NO)
```

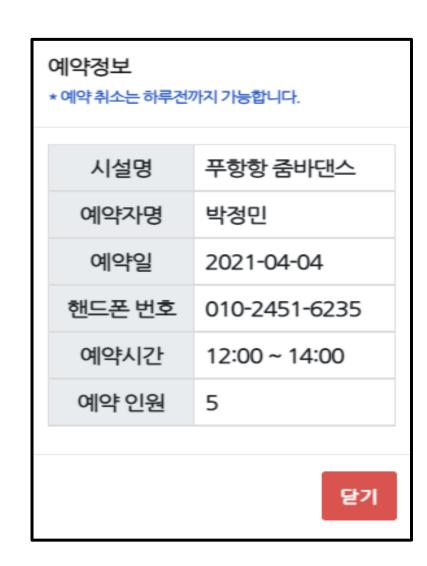
```
WHERE A.RESV CNT = B.CMNT CAPA
                      and RESV_START = p_resv_date||'T'||V_START_TIME||':'||'30'
                       and RESV_END = p_resv_date||'T'||LPAD(TO_NUMBER(V_START_TIME)+1,2,'0')||':'||'00';
 -- 30분 간격으로 인원을 조회한다.
                p_result := val; -- 실패라고 알려준다.
                EXIT WHEN val + p_resv_cnt > capa ; --인원이 다 차있으면 예약 등록 취소
                p_result := -1; -- 실패가 아니면 다시 참으로 결과 값을 변경해준다.
                 -- 반복획수 만큼 돌린다?
                select max(detail_no)+1 into detailNo from resvdetail;
                 INSERT INTO resvdetail (
                      detail_no, resv_no, detail_start, detail_end
                ) VALUES (
                      detailNo,p\_resv\_no,\ p\_resv\_date \|'T'\|V\_START\_TIME\|':'\|'30',\ p\_resv\_date \|'T'\|LPAD(TO\_NUMBER(V\_START\_TIME) + 1,2,'0')\|':'\|'00' + 1,2,'0'\|' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,2,'0' + 1,
                V_START_TIME := LPAD(TO_NUMBER(V_START_TIME)+1,2,'0'); -- 시작 시각을 1 증가시킨다.
                V_CHANGE := 0;
                SELECT NVL(MAX(A.RESV_CNT),0) AS RESV_CNT
                 FROM( SELECT
                                 B.CMNT NO
                                   , C.CMNT_NAME
                                   , DETAIL START AS RESV START
                                  , DETAIL_END AS RESV_END
                                   , SUM(RESV CNT) RESV CNT
                                  RESVDETAIL A LEFT OUTER JOIN RESERVATION B ON (A.RESV_NO = B.RESV_NO)
                                                    LEFT OUTER JOIN COMMUNITY C ON (B.CMNT_NO = C.CMNT_NO)
                            WHERE C.APT_CODE = P_APT_CODE
                               and c.cmnt no = P cmnt no
                            GROUP BY B.CMNT_NO, C.CMNT_NAME, DETAIL_START, DETAIL_END
                      ) A LEFT OUTER JOIN COMMUNITY B ON (A.CMNT_NO = B.CMNT_NO)
                 WHERE RESV_CNT = B.CMNT_CAPA
                       and RESV_START = p_resv_date||'T'||V_START_TIME||':'||'00'
                      and RESV_END = p_resv_date||'T'||V_START_TIME||':'||'30';
                  -- 30분 간격으로 인원을 조회한다
                 p_result := val; -- 실패라고 알려준다.
                 EXIT WHEN val + p_resv_cnt > capa ; --인원이 다 차있으면 예약 등록 취소
                p_result := -1; -- 실패가 아니면 다시 참으로 결과 값을 변경해준다.
                -- 반복획수 만큼 돌린다?
                select max(detail_no)+1 into detailNo from resvdetail;
                INSERT INTO resvdetail (
                      detail_no, resv_no, detail_start, detail_end
                      detailNo, p_resv_no, p_resv_date||'T'||V_START_TIME||':'||'00', p_resv_date||'T'||V_START_TIME||':'||'30'
                V CHANGE := 1;
          END IF;
           V_CNT := V_CNT+1;
     end loop;
```

# 시설 - 나의 예약 조회하기

### 입주민사이트 나의 예약



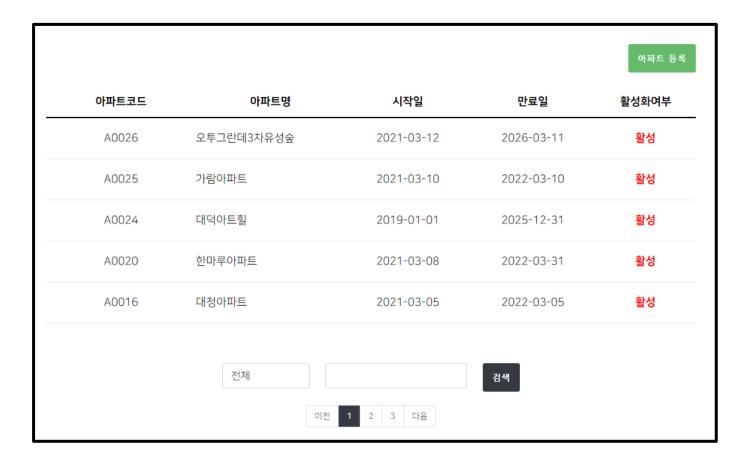
### 예약 상세 조회



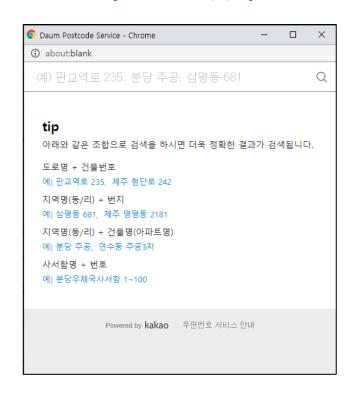
예약 정보 클릭 시 모달 창으로 예약 정보 조회 결과를 보여줍니다.

# 아파트 관리 – 목록, 등록

아파트 목록



우편번호 찾기



### 아파트 등록

우편번호 기본주소 상세주소 아파트이미지 난방정책 관리사무소 전화번	34159     우편번호 찾기       대전 유성구 학하중앙로 60       상세주소       파일 선택 선택된 파일 없음       지역난방
상세주소 아파트이미지 난방정책	상세주소 파일 선택 전택된 파일 없음
아파트이미지 난방정책	파일 선택 선택된 파일 없음
난방정책	
	지역난방
과리사민스 저하버	
호	010-5572-7833
관리소장명	이미정
계약시작일	2021-03-12
계약만료일	2026-03-11
	아파트 등록 시연용 입력

등록된 아파트 단지 리스트를 활성화 여부로 정렬하여 보여줍니다.

Daum 우편번호 API를 이용했습니다.

# 아파트 관리 – 상세보기, 수정

### 벤더 사이트 상세보기



### 벤더 사이트 정보수정



### 관리사무소 사이트 상세보기



# 아파트 관리 – 아파트 단지 생성

### 벤더 사이트 아파트 단지 생성

				돌아가기
동	Ď.	**************************************	면 적	단지생성
동	호	면적	삭제	•
1201	0301	98m²	삭제	
1201	0401	98m²	삭제	-
1201	0501	98m²	삭제	
1201	0601	98m²	삭제	
1201	0701	98m²	삭제	•

### 단지 생성 프로시저

```
create or replace PROCEDURE p_house(
   p_aptcode in apart.apt_code%type
  , p_dong in house.dong%type
  , p_floor in house.ho%type
  , p_ho in house.ho%type
  , p_area in HOUSE.HOUSE_AREA%type
   , p_result out number
  val number;
  v_cnt number := 1;
  out_cnt number :=0;
     select count(*)
         into val
      from house
      where \ instr(house\_code \ ,p\_aptcode||'D'||LPAD(p\_dong,4,'0')||'H'||to\_char(v\_cnt,'FM00')||'0'||p\_ho) \ > \ 0;
     if val = 0 then
         insert into house(
            house_code,
            dong,
            move_yn,
            apt_code,
            house_area
           p\_aptcode \|'D'\| LPAD(p\_dong,4,'0') \|'H'\| to\_char(v\_cnt,'FM00') \|'0'\| p\_ho
            , LPAD(p_dong,4,'0')
           , LPAD(to_char(v_cnt)||'0'||p_ho,4,'0')
            , P_APTCODE
            , p_area
         out_cnt := out_cnt+1;
     end if;
     exit when v_{cnt} = TO_NUMBER(p_floor);
     v_cnt := v_cnt+1;
   end loop;
   p_result := out_cnt;
```