

메이커 정보

메이커(기업)명:

사업자구분:

분리전화번호:

문의 이메일:

주소(사업장 주소):   
상세주소 입력해 주세요

사업자등록번호:

사업자등록증 사본:  선택한 파일 없음  
(resources/images/license\_copy.png, resources/images/)

메이커 프로필 이미지:  선택한 파일 없음  
(resources/images/default\_logo.png)

홈페이지 (선택사항):

카테고리 (선택사항):

**저장하기**

- Ajax를 통한 입력한 정보 DB에 저장 및 수정
  - 메이커 정보가 DB에 이미 저장되어 있는 경우에는 기존 데이터들을 새로 입력한 값으로 수정으로 해줌
  - 메이커 정보가 DB에 저장되지 않은 경우에는 새로 입력한 값을 DB에 저장함

```

/* 메이커 등록 */
@PostMapping("/insert")
public ResponseEntity<String> insertMaker(ModelAndView mv
    , Maker maker
    , HttpServletRequest req
    , HttpSession session
    , RedirectAttributes rttr
    , MultipartHttpServletRequest multipart
    ) {

    if (multipart.getFile("logo_file").getFileSize() > 0) {
        MultipartFile makerLogoFile = multipart.getFile("logo_file");
        String logoFile = fileUpload.saveFile(makerLogoFile, req);
        maker.setMaker_license_copy(logoFile);
    }

    if (multipart.getFile("license_copy_file").getFileSize() > 0) {
        MultipartFile makerLicenseCopyFile = multipart.getFile("license_copy_file");
        String licenseFile = fileUpload.saveFile(makerLicenseCopyFile, req);
        maker.setMaker_license_copy(licenseFile);
    }

    Member member = (Member) session.getAttribute("loginInfo");

    if (member != null) {
        maker.setEmail(member.getEmail());
    }

    try {
        String result = "";
        int i = makerService.insertMaker(maker);
        if (i == 1) {
            result = "success";
        } else {
            result = "fail";
        }
        return new ResponseEntity<>(result, HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.NOT_ACCEPTABLE);
    }
}

```

```

/* 메이커 수정등록 */
@PostMapping("/update")
public ResponseEntity<String> updateMaker(ModelAndView mv
    , Maker maker
    , HttpServletRequest req
    , HttpSession session
    , RedirectAttributes rttr
    , MultipartHttpServletRequest multipart
    ) {

    if (multipart.getFile("logo_file").getFileSize() > 0) {
        MultipartFile makerLogoFile = multipart.getFile("logo_file");
        String logoFile = fileUpload.saveFile(makerLogoFile, req);
        maker.setMaker_license_copy(logoFile);
    }

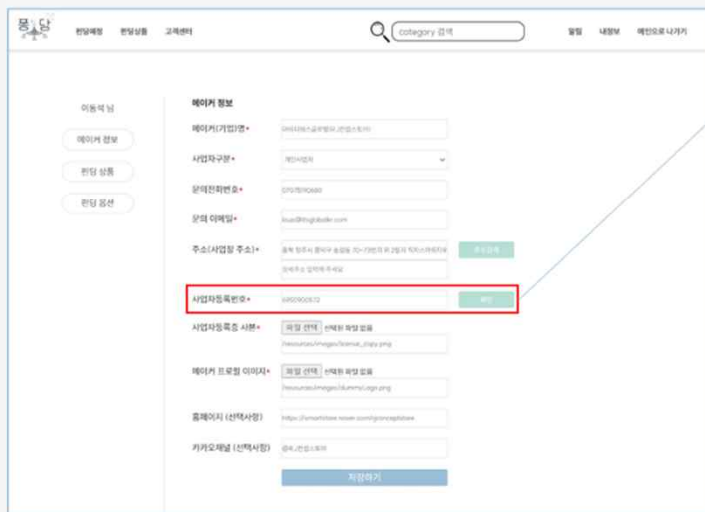
    if (multipart.getFile("license_copy_file").getFileSize() > 0) {
        MultipartFile makerLicenseCopyFile = multipart.getFile("license_copy_file");
        String licenseFile = fileUpload.saveFile(makerLicenseCopyFile, req);
        maker.setMaker_license_copy(licenseFile);
    }

    Member member = (Member) session.getAttribute("loginInfo");

    if (member != null) {
        maker.setEmail(member.getEmail());
    }

    try {
        String result = "";
        int i = makerService.updateMaker(maker);
        if (i == 1) {
            result = "success";
        } else {
            result = "fail";
        }
        return new ResponseEntity<>(result, HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.NOT_ACCEPTABLE);
    }
}

```



The screenshot shows a web form for '메이커 등록' (Maker Registration). The form includes fields for '메이커 정보' (Maker Information), '사업자등록' (Business Registration), '문의전화번호' (Inquiry Phone Number), '문의 이메일' (Inquiry Email), '주소' (Address), '사업자등록번호' (Business Registration Number), '사업자등록증 사진' (Business Registration Certificate Photo), '메이커 프로필 이미지' (Maker Profile Image), '홈페이지 (선택사항)' (Homepage (Optional)), and '카카오채널 (선택사항)' (Kakao Channel (Optional)). The '사업자등록번호' field is highlighted with a red box, and an arrow points from it to the JavaScript code block on the right.

```
//API 사업자등록 번호 확인
$("#license_button").click(function () {
    var form = {
        bNo: $("#b_no").val()
    }
    $.ajax({
        type: "POST"
        ,url: "<%=request.getContextPath()%>/maker/licenseCheck"
        ,data : form
        ,dataType:"JSON"
        ,success: function (result) {
            console.log("result : ", result);
            let str = result;
            let strArr = str.data;
            alert(strArr[0].tax_type);
            if (strArr[0].b_stt_cd != '') {
                $("#maker_register_num_yn").val("Y");
            }else{
                $("#maker_register_num_yn").val("N");
            }
        }
        ,error : function (result) {
            console.log(result);
        }
    });
});
$("#b_no").on("onchange", function() {
    $("#maker_register_num_yn").val("N");
});
```

```
/* 사업자등록번호 API */
@PostMapping("/licenseCheck")
public ResponseEntity<String> inserMaker(ModelAndView mv, HttpServletRequest req) {

    String url = "https://api.odcloud.kr/api/nts-businessman/v1/"
        + "status/servicekey=yokAhitJw0vyyvU9zRtniTovmD2Z18CR57jk85VqArcRRe%ZfduK2B1AgTt2BNt2FU75Xyn04NukTFd4qE4k5K2fPGRQK30K30";
    JSONObject jo = new JSONObject();
    JSONArray jarr = new JSONArray();

    jarr.put(req.getParameter("bNo"));
    jo.put("b_no", jarr);

    String result = apiRequestUtil.requestPost(url, jo.toString());

    try {
        return new ResponseEntity<>(result, HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.NOT_ACCEPTABLE);
    }
}
```

#### - 사업자등록 번호 확인

- 사업자등록번호를 조회하는 오픈 API를 사용하여 입력된 값의 진위여부를 확인하여 팝업 문구로 사용자에게 알려줌



```
<!-- sun start -->
<c:if test="${pick_yn eq 'N'}">
<button id="btn_like" type="button">
좋아요(<span id="pick_cnt">${pick_cnt}</span>)
</button>
</c:if>
<c:if test="${pick_yn eq 'Y'}">
<button id="btn_like" type="button">
싫어요(<span id="pick_cnt">${pick_cnt}</span>)
</button>
</c:if>
<input type="hidden" id="pick_yn" name="pick_yn" value="${pick_yn}" />
<input type="hidden" id="pick_update_yn" name="pick_update_yn" value="${pick_update_yn}" />
<input id="email" type="hidden" value="${loginInfo.email}">
<!-- sun end -->
```

- 펀딩 상세페이지로 이동 시, 좋아요 여부를 mv에 담아서 이동
  - 좋아요를 한 경우에는 좋아요 옆의 사진이 색이 채워진 하트로 표시됨  
=> pickYn = 'N'
  - 좋아요를 하지 않은 경우에는 좋아요 옆의 사진이 색이 채워지지 않은 하트로 표시됨  
=> pickYn = 'Y'

4일 남음

2022.08.04 종료

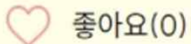
달성률 66% 목표금액 1,000,000원

누적 펀딩액 658,000원

서포터 10명



펀딩하기



종아요(0)



공유하기



판매자에게 문의하기



신고하기

```
$("#btn_like").on('click', function(){
    if($("#email").val() == ''){
        alert('로그인 해주세요.');
```

```
// pick insert
@PostMapping("/insert")
public ResponseEntity<String> insertPick(Pick pick) {
    try {
        pickService.insertPick(pick);
        JSONObject jo = new JSONObject();
        jo.put("pick_yn", "Y");
        jo.put("pick_cnt", pickService.countPick(pick));
        jo.put("pick_update_yn", "Y");
        String result = jo.toString();
        return new ResponseEntity<>(result, HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.NOT_ACCEPTABLE);
    }
}
```

- Ajax를 통한 데이터 DB에 저장
  - 종아요 업데이트 여부 값을 확인하여 종아요 값을 저장함
  - email, p\_no, pickYn 값을 json 타입으로 변경하여 controller에 전달하고 DB에 저장함
    - => 전달된 값을 DB에 저장하고 종아요 여부, 종아요 수, 종아요 업데이트 여부를 리턴함
  - 종아요 사진이 색이 채워진 하트로 변경되고, 종아요 한 사람 수가 1증가함

**4일 남음** 2022.08.04 종료  
달성률 66% 목표금액 1,000,000원  
누적 펀딩액 658,000원  
서포터 10명



펀딩하기



좋아요(1)



공유하기



판매자에게 문의하기



신고하기

```

$("#btn_like").on('click', function(){
    if($("#email").val() == ''){
        alert('로그인 해주세요.');
```

 }
 return;

 var urlStr;
 if (\$("#pick\_update\_yn").val() == 'Y') {
 urlStr = "<%=request.getContextPath()%>/pick/update";
 } else {
 urlStr = "<%=request.getContextPath()%>/pick/insert";
 }

 var pickYn = \$("#pick\_yn").val();
 if (pickYn == 'N') {
 pickYn = 'Y';
 } else {
 pickYn = 'N';
 }

 \$.ajax({
 url: urlStr,
 type: "post",
 data: {
 email: \$("#email").val(),
 p\_no: \$("#p\_no").val(),
 pick\_yn: pickYn
 },
 datatype: "JSON",
 success: function(result){
 console.log(result);
 var data = JSON.parse(result);
 console.log(data);
 if (data.pick\_yn == 'Y') {
 \$("#pick\_yn").val('Y');
 var src = "<%=request.getContextPath()%>/resources/images/heart\_color.png";
 \$("#pick\_img").attr('src', src);
 } else {
 var src = "<%=request.getContextPath()%>/resources/images/heart.png";
 \$("#pick\_yn").val('N');
 \$("#pick\_img").attr('src', src);
 }
 \$("#pick\_cnt").text(data.pick\_cnt);
 \$("#pick\_update\_yn").val(data.pick\_update\_yn);
 },
 error: function(request, status, error) {
 console.log("code:" + request.status + "\n" + "message:" + request.responseText + "\n" + "error:" + error);
 }
 });
});
 

```

// pick delete
@PostMapping("/update")
public ResponseEntity<String> updatePick(Pick pick){
    try {
        pickService.updatePick(pick);
        JSONObject jo = new JSONObject();
        jo.put("pick_yn", pick.getPick_yn());
        jo.put("pick_cnt", pickService.countPick(pick));
        jo.put("pick_update_yn", "Y");
        String result = jo.toString();
        return new ResponseEntity<>(result, HttpStatus.OK);
    } catch (Exception e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.NOT_ACCEPTABLE);
    }
}
    
```

- Ajax를 통한 데이터 DB에서 수정
  - 좋아요 업데이트 여부 값을 확인하여 좋아요 값을 수정함
  - email, p\_no, pickYn 값을 json 타입으로 변경하여 controller에 전달하고 DB에서 수정함  
=> 전달된 값을 DB에 저장하고 좋아요 여부, 좋아요 수, 좋아요 업데이트 여부를 리턴함
  - 좋아요 사진이 색이 비워진 하트로 변경되고, 좋아요 한 사람 수가 1감소함