# 프로젝트 핵심 기술 소개서

**[ BODY BUDDY : 피트니스 통합 자원관리 플랫폼 ]**

**TEAM FM**


**구성원 :** { 고정근, 권순철, 박한솔, 윤상기, 장기림 }


**개발 일정 :** { 2019.5.8 ~ 2019.6.28 (42 일) }

# INDEX


1. 프로젝트 개요 3
2. 메인 페이지 출력 4
3. 광고 상세페이지 보기 9
4. 회원 가입 17
5. - 아이디 중복체크 19
6. - 이메일 인증 22
7. 로그인 25
8. - 아이디 찾기 27
9. 실시간 알림 기능 29
10. 프로그램 찜하기 34
11. 프로그램 찜 취소하기 36
12. 구매하기, 옵션 만료기능(정원) 38
13. 옵션과 광고 만료기능(기간) 42

**<업체, 트레이너 공통 관리기능>**

14. -광고 글 쓰기 43
15. -광고 수정Form 가져오기 50
16. -광고 수정하기 52
17. -광고 삭제하기 57
18. -관리페이지 메인 대시보드 59
19. -프로그램 별 회원 출결 관리 67
20. -회원 이름 검색 71
21. -문의 답변 등록하기 73

**<트레이너>**

22. -소속 업체 검색 77
23. -업체 소속 요청 79
24. -업체 소속 취소 81
25. -프로필 수정하기 82
26. -트레이너 정보 수정 84

**<업체>**

27. -일반프로그램 이용 상태 전환 86
28. -트레이너 소속요청 수락/거절 88
29. -트레이너 소속 끊기 91
30. -트레이너 근태 상태 입력 93
31. -업체 정보 수정페이지 이동 95
32. -업체 정보 수정 97

**<일반회원>**

33. -내 프로그램 보기 100
34. -내 출석 보기 102
35. -내 후기/문의 목록 104
36. -후기 글쓰기 페이지 106
37. -후기 글 등록 108
38. -내 찜 목록 내역 보기 109
39. -내 결제 내역 보기 110
40. -내 정보보기 페이지 이동 111
41. -내 정보 수정 113
42. 프로필 보기 117

## 1. BODY BUDDY (피트니스 통합 자원관리 플랫폼)

### ■ 개요

1. 지역 및 카테고리 별 업체와 트레이너 홍보와 위치정보 제공
2. (일반 소비자 회원) – (업체 회원) – (소속/개인 트레이너 회원) 사이 상호작용 및 자원관리/조회 가능

### ■ 개발 규모

**1. DB**

1) DB Table : 17 개 (총 Column : 84 개)
2) 사용된 Query : 152 개

**2. Source**

1) Class (개인권한별 분류)
   - Controller : 6 개
   - URL - 89 개
   - Bean/DTO : 12 개
   - Service : 6 개
   - DAO : 6 개
   - Method : 212 개
2) View (JSP) : 51 개

### ■ 사용 기술

**1. Tools** : STS 3.9.8, Brackets, SQL Developer
**2. Language** : Java, JavaScript, JSP, HTML, CSS
**3. Framework** & **Libraries**: Spring, Spring WebSocket(SockJS), Spring Security, Mybatis, JQuery-Ajax, Bootstrap, Maven, JSTL, Gson, Jackson
**4. DB** : ORACLE **-Modeling** : eXERD, Microsoft Excel
**5. Server** : Tomcat v9.0
**6. Design Pattern** : MVC2
**7. External APIs :** Google Mail, Chart.js, DaumKakao Address & Map, CKEditor, BootStrap - Majestic Admin
**8. Configuration Management**: GitHub – Atlassian-SourceTree

## 2 JOB_NAME : 메인 페이지 출력 작성자 : 장기림

### ■ 개요

1. 메인 광고리스트 출력.
2. 카테고리 유무, 지역(도로명/지번주소) 유무 등 모든 경우의 수에 따라 리스트 출력 및 지도 표시.
3. 지도에 마커 표시 및 인포윈도우 상호작용기능 구현
4. 일반 회원 로그인 했을 경우 찜 여부에 따라 찜 버튼이 달라져야 함.
5. 업체 회원의 경우 업체 상세주소/ 개인 트레이너의 경우 트레이너의 약식 주소 출력

### ■ CONTROLLER

```
@RequestMapping(value = "/")
public ModelAndView mainList(String sido, String sigungu, String extra, String cate) {
  mav=ys.mainList(sido, sigungu, extra, cate); //시도, 시군구, 상세주소, 카테고리
  return mav;
}
```

### ■ SERVICE

카테고리 확인 ➡ 주소지 확인 ➡ 도로명/지번 확인 ➡ 찜리스트 출력 ➡ makeHTML

```
public ModelAndView mainList(String sido, String sigungu, String extra, String cate) {
        mav=new ModelAndView();
        String view=null;
        List<Map<String, String>> mainList=null; //메인리스트
        List<Map<String, String>> dibsList=null; //찜리스트
        mav.addObject("showMap",false); //지도 표시여부
        if(sido!=null) {//주소지가 있는경우 : 주소지 기준 메인리스트 출력
                Map<String,String> local = new HashMap<>();
                local.put("sido", sido);
                local.put("sigungu", sigungu);
                if(cate!=null) { //주소지 있는경우 -> cate있는경우: 카테고리기준 메인리스트 출력
                        local.put("cate", cate);
                }
                if(extra.contains("/")){
                String[] extraSplit = extra.split("/");//도로명주소/지번주소 나눔
                        local.put("extra1", extraSplit[0]);
                        local.put("extra2", extraSplit[1]);
                        mainList = yDao.mainListMap(local);
                }else {
                        local.put("extra", extra);
                        mainList = yDao.mainListMap(local);
                }
                mav.addObject("showMap",true); //지도 표시
                mav.addAllObjects(local);
        }else {//주소지가 없는경우
                if(cate!=null) {//주소지 없는경우 -> cate있는경우
                        mainList = yDao.mainListCate(cate);
                }else {//주소지 cate 둘 다 없는 경우
                        mainList = yDao.mainList();
                }
        }
        // 로그인 되어있을 경우 : 찜 리스트 출력
        Member sessionMb = (Member) session.getAttribute("mb");
        if (sessionMb != null) {
                String d_id = sessionMb.getM_id();
                dibsList = yDao.dibsN(d_id);
        }
```

```
                String html = makeHTMLMainList(mainList, dibsList, sessionMb, mav); //makeHTML
                mav.addObject("mainListHTML",html);
                String jsonMainList = new Gson().toJson(mainList);
                mav.addObject("jsonMainList", jsonMainList);
                view="main";
                mav.setViewName(view);
                return mav;
        }
private String makeHTMLMainList(List<Map<String, String>> mainList, List<Map<String, String>>
dibsList,
                        Member sessionMb, ModelAndView mav) {
  StringBuilder sb = new StringBuilder();
  sb.append("<div id='listCard' class=\"col-md-12 card scroll\">" +
            "    <div class=\"card\">" +
            "        <div class=\"card-body\">" +
            "            <p class=\"card-title\">총"+mainList.size()+"건의 결과가 있습니다.</p>" +
            "            <div class=\"row\">");
  Set<String> delBtnSet = new HashSet<>();
  String addBtn=null;
  String delBtn=null;
  for(int i=0; i<mainList.size();i++) {
    String ad_code = mainList.get(i).get("AD_CODE").toString();
    String ad_category = mainList.get(i).get("AD_CATEGORY").toString();
    sb.append("            <div class=\"adBanner col-sm-6 col-md-3\">" +
              "              <div class=\"thumbnail\">" +
              "                <img src='resources/upload/"+mainList.get(i).get("PF_IMAGE")+"'"+
                                    "data-holder-rendered=\"true\" style=\"height: 200px; width: 100%;
                                     display: block;\">"+
              "                <div class=\"caption\"><br>" +
              "                  <h3 id=\"thumbnail-label\">" +
                                  mainList.get(i).get("AD_TITLE")+
              "                  <a class=\"anchorjs-link\" href=\"#thumbnail-label\">"+
              "                  <span class=\"anchorjs-icon\"></span></a></h3>");
    //주소 출력 //소속업체가 없는경우
    if(mainList.get(i).get("T_CID")==null || mainList.get(i).get("T_CID").equals("소속업체없음")) {
          sb.append("<p>"+mainList.get(i).get("M_ADDR")+"</p>");
    }else {//소속업체 있는경우
          sb.append("<p>"+mainList.get(i).get("T_CID_ADDR")+"</p>");
    }//주소 출력끝
    //카테고리 출력
    sb.append("<p>");
    if(ad_category.equals("일반")) {
          sb.append("일반 정기 회원 모집");
    }else {
          sb.append(ad_category);
    }
    sb.append("</p>" +
    //카테고리 출력 끝
              "                <p><a href='detailpage?ad_code="+ad_code+
                                  "'id='showdetail"+ad_code+"' class='btn btn-primary'
                                  role=\"button\">상세보기</a> ");
    //찜버튼 출력
    delBtn = "<button id='dibsDelete"+ad_code+"'type=\"button\" class=\"btn btn-outline-danger btn-
rounded btn-icon\"><i class=\"mdi mdi-heart\"></i></button>";
    addBtn = "<button id='dibsAdd"+ad_code"'type=\"button\" class=\"btn btn-outline-secondary btn-
rounded btn-icon\"><i class=\"mdi mdi-heart-outline text-danger\"></i></button>";
    //(회원:찜하지 않은 상품은 찜하기버튼)
    if(sessionMb != null && (dibsList != null && dibsList.size() != 0)) {
          for (int j = 0; j < dibsList.size(); j++) {
            if(dibsList.get(j).get("D_ADCODE").equals(ad_code)) {
              delBtnSet.add(ad_code);
            }
          }
    //회원인데 찜 하나도 없을때도 dibsList null일수있음 : 찜하기버튼
    }else if(sessionMb == null && (dibsList == null || dibsList.size() == 0)) {
```

```
            //비회원 세션에 찜한상품 아니면 찜하기버튼
            if(session.getAttribute("tempDibs" + ad_code)==null
              ||session.getAttribute("tempDibs" + ad_code) != "dibs") {
                sb.append(addBtn);
             //비회원 세션에 찜한 상품 : 찜취소버튼
             }else if(session.getAttribute("tempDibs" + ad_code) == "dibs") {
                sb.append(delBtn);
             }
        }// 찜버튼 끝
        sb.append("</p>" +
                "                                        </div>" +
                "                                      </div>" +
                "                                    </div>"
        );
      }//end For
      String delBtnSetJson = new Gson().toJson(delBtnSet);
      mav.addObject("delBtnSet",delBtnSetJson);
      sb.append("                                  </div>" +
                "                                </div>" +
                "                              </div>" +
                "                            </div>");
      return sb.toString();
    }
```

(업체/트레이너의 각각 주소 출력, 카테고리 출력, 찜 출력에 따라 경우의 수 고려)

## ■ DAO & MAPPER

```
List<Map<String, String>> mainList();
<select id="mainList" resultType="HashMap">
        SELECT DISTINCT * FROM MAINLIST
        LEFT JOIN T_CID_ADDR ON MAINLIST.T_CID=T_CID_ADDR.T_CID
        ORDER BY AD_CODE DESC
</select>

List<Map<String, String>> mainListCate(String cate);
<select id="mainListCate" parameterType="String" resultType="HashMap">
        SELECT DISTINCT *
        FROM MAINLIST
        LEFT JOIN T_CID_ADDR ON MAINLIST.T_CID=T_CID_ADDR.T_CID
        WHERE AD_CATEGORY= #{cate}
        ORDER BY AD_CODE DESC
</select>
```

(myBatis 동적SQL을 사용해 도로명주소/지번주소 구분해서 Query)

```
List<Map<String, String>> mainListMap(Map<String, String> local);
<select id="mainListMap" parameterType="HashMap" resultType="HashMap">
        <![CDATA[
                SELECT DISTINCT * FROM MAINLIST
                LEFT JOIN T_CID_ADDR ON MAINLIST.T_CID=T_CID_ADDR.T_CID
                WHERE
        ]]>
        <if test="cate!=null">
                AD_CATEGORY= #{cate} AND
        </if>
        <if test="extra1!=null or extra2!=null">
                <![CDATA[
                        ((M_ADDR LIKE '%${sido}%' AND M_ADDR LIKE '%${sigungu}%'
                          OR M_ADDR LIKE '%${extra1}%' OR M_ADDR LIKE '%${extra2}%')
                          OR (T_CID_ADDR LIKE '%${sido}%' AND T_CID_ADDR LIKE '%${sigungu}%'
                          OR T_CID_ADDR LIKE '%${extra1}%' OR T_CID_ADDR LIKE '%${extra2}%'))
                ]]>
        </if>
        <if test="extra!=null">
                <![CDATA[
                        ((M_ADDR LIKE '%${sido}%' AND M_ADDR LIKE '%${sigungu}%'
                          OR M_ADDR LIKE '%${extra}%')
                          OR (T_CID_ADDR LIKE '%${sido}%' AND T_CID_ADDR LIKE '%${sigungu}%'
                          OR T_CID_ADDR LIKE '%${extra}%'))
```

```
                ]]>
        </if>
        ORDER BY AD_CODE DESC
</select>
```

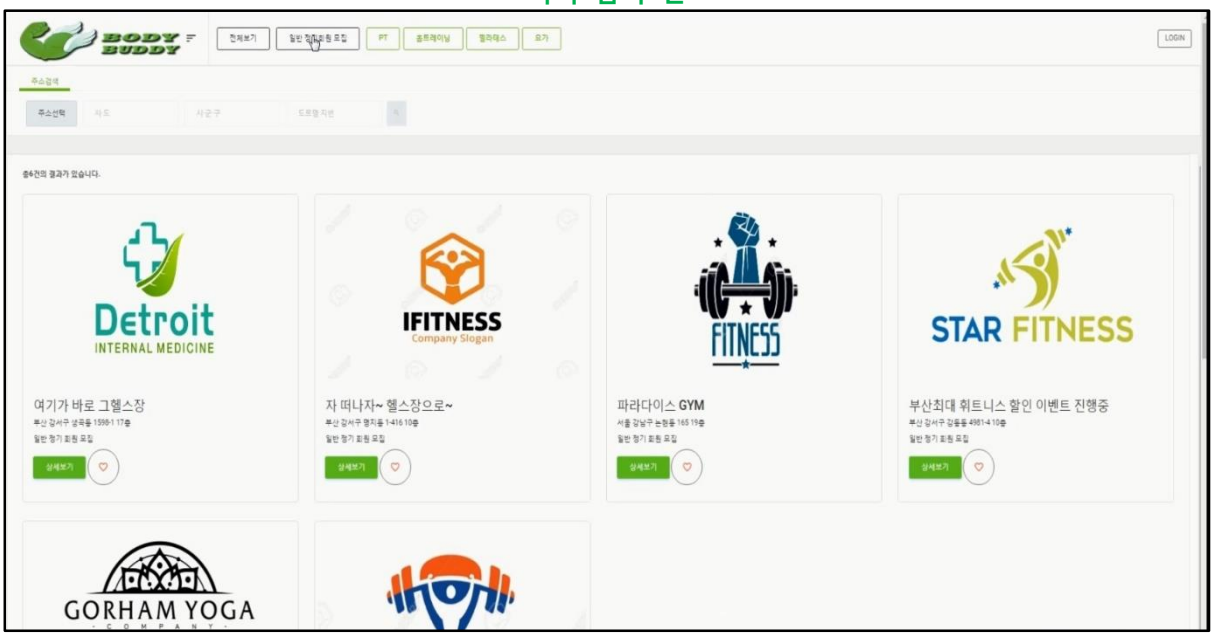
## ■ FRONT-END

```
var cp = "${pageContext.request.contextPath}";//컨텍스트 패스
var jsonMainList = "${jsonMainList}"; //메인 광고 리스트
var addrList = new Array();//주소 리스트 편집
for(var i=0;i<jsonMainList.length; i++){
        var addrRecord = jsonMainList[i]["M_ADDR"];
        addrList[i] += addrRecord.replace("undefined","");
}
for(var i=0;i<addrList.length; i++){
        addrList[i]=addrList[i].replace("undefined","");
}
for(var i=0; i< addrList.length; i++){ //주소리스트 검색 및 지도작업
    geocoder.addressSearch(addrList[i], function(results, status) {//검색
        if(status === daum.maps.services.Status.OK) {//검색결과가 있다면
            var result = results[0]; //첫번째 결과의 값을 활용
            var markerPosition  = new daum.maps.LatLng(result.y, result.x);// 마커 위치좌표
            var marker = new daum.maps.Marker({// 마커 생성
              position: markerPosition
            });
            marker.setMap(map);// 마커표시
            map.setCenter(markerPosition); // 지도 중심 변경
            i--;
            var infowindow = new kakao.maps.InfoWindow({ // 마커에 인포윈도우 생성
              content: "<img alt='pfimageList[i]' src='"+cp+"/resources/upload/"
                    +jsonMainList[i]["PF_IMAGE"]+"' width='100px'><br><h4 class='text-success'>"
                    +jsonMainList[i]["AD_TITLE"]+"</h4><br>"
                    +jsonMainList[i]["M_ID"] // 인포윈도우에 표시할 사진+광고 제목+아이디
            });
          (function(marker, infowindow) {//인포윈도우 컨트롤
                    kakao.maps.event.addListener(marker, 'mouseover', function() {
                      infowindow.open(map, marker);
                    });
                    kakao.maps.event.addListener(marker, 'mouseout', function() {
                      infowindow.close();
              });
          })(marker, infowindow);
      }//if END
   });//geoCoder END
}//for END
var showMap = ${showMap} //지도 보여주기 여부
if(showMap){
        $(".adBanner").prop("className","adBanner col-sm-6 col-md-4");
        $("#listCard").prop("className","col-md-8 card scroll");//col-md-12 card scroll
        $("#mapCard").prop("hidden",false);
        mapContainer.style.display = "block";
        map.relayout();
}
```

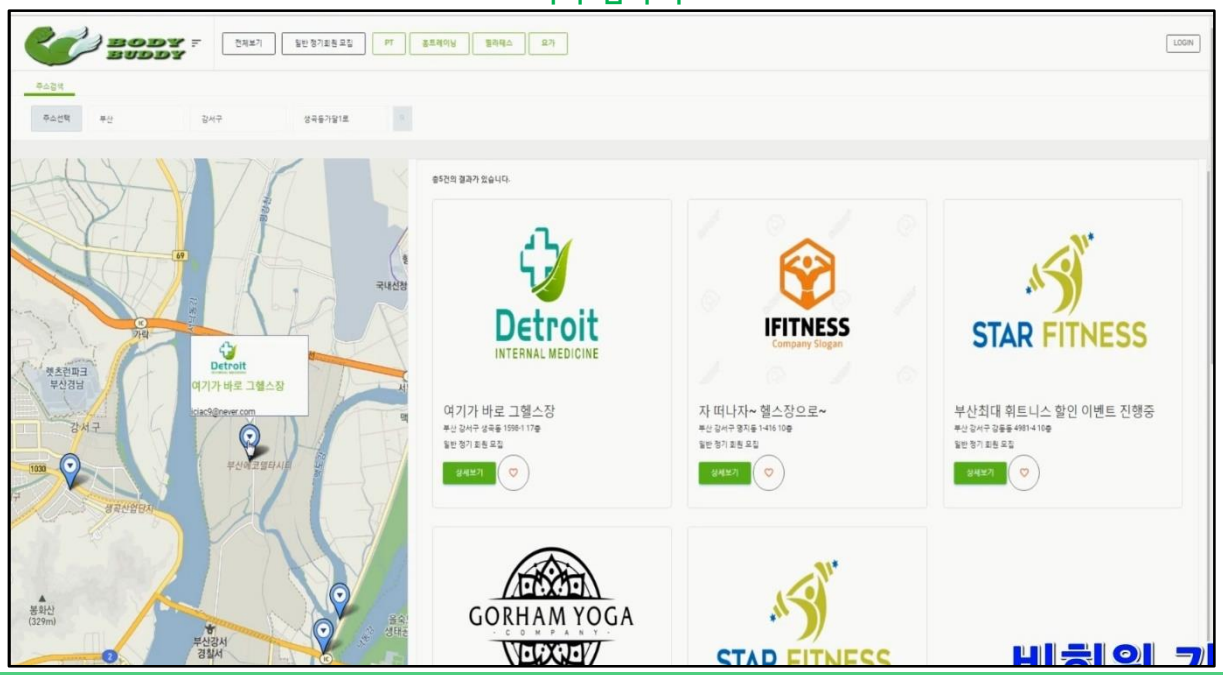
(외부 API 를 활용한 지도 및 마커와 인포윈도우 출력)

## ■ 결과

-지역 검색 전-




-지역 검색 후-



## 3 JOB_NAME : 광고 상세페이지 보기 작성자 : 장기림

1. 메인리스트에 있는 상세보기 버튼 클릭 시 이동되는 상세정보 페이지
2. 상세 보기(기본값), 별점 및 후기 보기, 문의 보기 탭으로 구성
3. 광고사진 부트스트랩 Carousel 활용
4. 회원 종류 및 상품 조건, 옵션별 접근유효성 검사 구현
   - 찜: 비회원, 일반회원 - 구매: 일반회원 - 트레이너 프로필 보기: 프로그램 상품
   - 개인 주소/업체 주소 : 트레이너 프로그램/업체 프로그램
5. 별점의 평점 계산 및 후기 내용 더보기/접기 버튼 구현

### ■ CONTROLLER

```
@RequestMapping(value = "/detailpage", method = RequestMethod.GET)
public ModelAndView detailPage(String ad_code) {
        System.out.println("url:/detailpage");
        mav = ks.detailPage(ad_code);
        return mav;
}
```

### ■ SERVICE

필요한 정보들 → makeHTML → 광고사진 → 주소 → 옵션 → 찜/구매/트레이너 프로필 → 상세보기/별점,후기/문의

```
public ModelAndView detailPage(String ad_code) {
                mav = new ModelAndView();
                String view = null;
                Member sessionMb=null;
                List<Map<String, String>> dibsList=null;

                Map<String, String> dp = kDao.detailPage(ad_code);//상세정보
                List<OpCategory> opCateList = kDao.opCateList(ad_code);//카테고리 정보
                List<Map<String,String>> apList =kDao.adPhotoList(ad_code);//광고 사진리스트
                List<Qna> qaList = kDao.detailQa(ad_code);//문의 리스트
                List<Map<String,String>> rvList = kDao.detailReview(ad_code);//후기 리스트
                List<Map<String,String>> psList = kDao.detailPsCount(ad_code);//옵션 구매 수량
                sessionMb = (Member) session.getAttribute("mb");

                if(sessionMb!=null) {
                        String d_id = sessionMb.getM_id();
                        dibsList = yDao.dibsN(d_id);//찜 유무 판단
                }
                String html = makeHTMLDetailPage(dp, opCateList, dibsList, sessionMb,
                                                qaList, rvList, psList, mav, apList);
                view = "detailPage";
                mav.addObject("detailPageHTML", html);
                mav.setViewName(view);
                return mav;
        }

private String makeHTMLDetailPage(Map<String, String> dp, List<OpCategory> opCateList,
                                List<Map<String, String>> dibsList, Member sessionMb,
                                List<Qna> qaList, List<Map<String, String>> rvList,
                                List<Map<String, String>> psList, ModelAndView mav,
                                List<Map<String, String>> apList) {
  StringBuilder sb = new StringBuilder();
  String ad_code = dp.get("AD_CODE").toString();
  sb.append("<div class=\"content-wrapper\">" +
            "<div class=\"row\">" +
             "<div class=\"col-md-5 grid-margin stretch-card\">" +
```

```
                "<div class=\"card\">" +
                 "<div class=\"card-body\">" +
                  "<div class=\"col-md-12\" style=\"overflow: hidden; height: 400px;\">" +
//광고사진 시작
                   "<div id=\"carousel-adPhoto-generic\" class=\"carousel slide\"
                    data-ride=\"carousel\" data-interval=\"false\">" +
                    "<ol class=\"carousel-indicators\">");
                    for(int i=0;i<apList.size();i++) {
                    sb.append("<li data-target=\"#carousel-adPhoto-generic\" data-slide-to='"+i+"'");
                   if(i==0) {
                    sb.append("class='active'");
                   }
                   sb.append("></li>");
                  }//END for
                  sb.append("</ol><div class=\"carousel-inner\" role=\"listbox\">");
                  for(int i=0;i<apList.size();i++) {
                   sb.append("<div class='item");
                   if(i==0) {
                    sb.append(" active");
                   }
                   sb.append("')><img src='resources/upload/"+apList.get(i).get("AP_IMAGE")+"'>
                             "</div>");
                  }//END for
                  sb.append("</div>" +
                           "<a class='left carousel-control' href='#carousel-adPhoto-generic' role='button' data-
                                 slide='prev'>" +
                           "<span class='glyphicon glyphicon-chevron-left'
                            aria-hidden=\"true\"></span>" +
                           "<span class=\"sr-only\">Previous</span>" +
                           "</a>" +
                           "<a class='right carousel-control' href='#carousel-adPhoto-generic'
                                 role=\"button\" data-slide=\"next\">" +
                           "<span class=\"glyphicon glyphicon-chevron-right\"
                            aria-hidden=\"true\"></span>" +
                           "<span class=\"sr-only\">Next</span>" +
                           "</a>" +
                         "</div>"+
//광고사진 캐러셀 끝
                     "</div>" +
                    "</div>" +
                   "</div>" +
                  "</div>" +
                  "<div class=\"col-md-6 grid-margin stretch-card\">" +
                   "<div class=\"card\">" +
                    "<div class=\"card-body\">" +
                     "<input type=\"hidden\" id=\"ad_code\" name=\"ad_code\" value='"+ad_code+"'>"+
                     "<input type=\"hidden\" id=\"ad_name\" name=\"ad_name\"
                                                                 value='"+dp.get("AD_NAME")+"'>"+
                     "<div class=\"caption\">" +
                      "<h3 style=\"text-align:center\">"+dp.get("AD_TITLE")+"<br>"+
                       "<small class=\"text-muted\">");
//주소 시작위치 //소속업체가 없는경우
                  if(dp.get("T_CID")==null) {
                           sb.append(dp.get("M_ADDR"));
                  }else {//소속업체 있는경우
                           sb.append(dp.get("T_CID_ADDR"));
                  }
//주소 끝위치
                  sb.append("</small></h3><br>" +
                            "<h5>"+dp.get("AD_CATEGORY")+"</h5><br>" +
//옵션출력 시작
                     "<p><select id='optionSelect' class=\"form-control\">" +
                     "<option id='opSelPlz'>상세 옵션을 선택해주세요</option>");
                  for(int i=0;i<opCateList.size();i++) {//옵션명 찍어주기
                   sb.append("<option id='op"+opCateList.get(i).getOp_code()+"'>"+
                              opCateList.get(i).getOp_content()+" - ");
                   if(opCateList.get(i).getOp_period()!=null)
```

```
                    sb.append("기간 : "+opCateList.get(i).getOp_period()+"-");
                   if(opCateList.get(i).getOp_times()!=null)
                    sb.append("횟수 : "+opCateList.get(i).getOp_times()+"-");
                   if(opCateList.get(i).getOp_day()!=null)
                    sb.append("요일 : "+opCateList.get(i).getOp_day()+"-");
                   if(opCateList.get(i).getOp_clock()!=null)
                    sb.append("시간 : "+opCateList.get(i).getOp_clock()+"-");


                  if(opCateList.get(i).getOp_personnel()!=-100) {//정원이 무제한이 아닌경우
                    sb.append("남은 인원/정원 : ");//남은 자리/정원 뽑아내기
                    int cnt=0;
                    int leftCnt=0;
                    if(psList.size()!=0) {
                     for(int j=0; j<psList.size();j++) {
                      if(opCateList.get(i).getOp_code().equals(psList.get(j).get("PS_OPCODE"))) {
                       cnt++;
                      }
                      leftCnt = opCateList.get(i).getOp_personnel()-cnt;
                     }//END for
                    }else if(psList.size()==0){
                     leftCnt=opCateList.get(i).getOp_personnel();
                    }
                    sb.append(leftCnt+"/"+opCateList.get(i).getOp_personnel()+"</option>");
                  }else if(opCateList.get(i).getOp_personnel()==-100){//정원이 무제한일경우
                   sb.append("인원 제한 없음</option>");
                  }
                 }//END for
                 sb.append("</select></p>");
//옵션출력 끝
//옵션코드 리스트, 옵션가격 리스트 뽑아내기
                 for(int i=0;i<opCateList.size();i++) {
                  sb.append("<input type=\"hidden\" id='"+opCateList.get(i).getOp_code()+"'
                              value='"+opCateList.get(i).getOp_price()+"'/>");
                 }
                 sb.append("<div class=\"row\">" +
                            "<div class=\"col-md-7\">" +
                             "<blockquote class=\"blockquote blockquote-primary\">" +
                              "<footer class=\"blockquote-footer\">프로모션 및 가격 할인행사</footer>" +
                              "<p>PT100회 한정수량 골라잡기 이벤트</p>" +
                             "</blockquote>" +
                            "</div>" +
                            "<div class=\"col-md-4\">" +
                             "<blockquote  id='priceShow' class=\"blockquote blockquote-primary\">" +
                              "<h3><small class=\"text-muted\">가격: </small>0원</h3>" +
                             "</blockquote>" +
                            "</div>" +
                           "</div>"+
                           "<div style='text-align: center'>");
//버튼단DIV 시작
//옵션 담당 트레이너 보기버튼 시작
                if(!dp.get("AD_CATEGORY").toString().equals("일반")) {
                     sb.append("<button type='button' class='btn btn-outline-secondary dropdown-toggle'
                                 data-toggle='dropdown'>담당 트레이너 프로필 보기</button>" +
                              "<div class='dropdown-menu'>");
                     //프로그램 광고 작성자가 트레이너
                   if(dp.get("M_KIND").toString().equals("t") ) {
                     sb.append("<a href='#' id='profilePage"+dp.get("AD_NAME")+
                                "'class='dropdown-item profilePage' data-toggle='modal'
                                 data-target='#myModal'>"+dp.get("M_NAME")+
                           "</a>");
```

```
                    //프로그램 광고 작성자가 업체
                  }else if(dp.get("M_KIND").toString().equals("c")) {
                     for(int i=0;i<opCateList.size();i++) {
sb.append("<a href='#' id='profilePage"+opCateList.get(i).getOp_trainer()+
                              "'class='dropdown-item profilePage' data-toggle='modal'
                                data-target='#myModal'>"+opCateList.get(i).getM_name()+
                            "</a>");
                     }
                  }
               }
          sb.append("</div>");
//옵션 담당 트레이너 보기버튼 끝
//찜버튼 위치
          Set<String> delBtnSet = new HashSet<>();
          String addBtn=null;
          String delBtn=null;
          addBtn = "<button id='dibsAdd"+ad_code+"'type='button' class='btn btn-outline-secondary
                 btn-rounded btn-icon'>"+"<i class=\"mdi mdi-heart-outline text-danger\"></i></button>";
          delBtn = "<button id='dibsDelete"+ad_code+"'type='button' class='btn btn-outline-danger
                 btn-rounded btn-icon'>"+"<i class=\"mdi mdi-heart\"></i></button>";
          if (sessionMb != null && (dibsList != null && dibsList.size() != 0)) {
            for (int j = 0; j < dibsList.size(); j++) {
              if(dibsList.get(j).get("D_ADCODE").equals(ad_code)) {
                delBtnSet.add(ad_code);
              }
            }
          }else if(sessionMb == null && (dibsList == null || dibsList.size() == 0)) {                          if
(session.getAttribute("tempDibs" + ad_code) == null
                                                || session.getAttribute("tempDibs" + ad_code) != "dibs") {
              sb.append(addBtn);
            }else if (session.getAttribute("tempDibs" + ad_code) == "dibs") {
              sb.append(delBtn);
            }
          }
//찜버튼 끝
          String delBtnSetJson = new Gson().toJson(delBtnSet);
          mav.addObject("delBtnSet",delBtnSetJson);
          sb.append("<button id='purchase' class='btn btn-primary'
                      role='button' disabled='true'>구매</button>
                    </div>" +
                  "</div>" +
//버튼단 끝
                  "      </div>" +
                  "    </div>" +
                  "  </div>" +
                  "</div>" +
                  "<div class=\"row\">" +
                  "  <div class=\"col-md-11 grid-margin stretch-card\">" +
                  "    <div class=\"card\">" +
                  "      <div class=\"card-body dashboard-tabs p-0\">" +
                  "        <ul class=\"nav nav-tabs px-4 nav-justified\" role=\"tablist\">" +
                  "          <li class=\"nav-item\">" +
                  "            <a class=\"nav-link active\" id=\"detailInfo-tab\"
                                 data-toggle=\"tab\" href=\"#detailInfo\" role=\"tab\"
                                 aria-controls=\"detailInfo\" aria-selected=\"true\"
                                 style=\"border-bottom-color : #71c016; color: #71c016\">
                                 상세 정보 보기</a>" +
                  "          </li>" +
                  "          <li class=\"nav-item\">" +
                  "            <a class=\"nav-link\" id=\"review-tab\" data-toggle='tab' href='#review'
                                 role=\"tab\" aria-controls=\"review\" aria-selected=\"false\"
                                 style=\"border-bottom-color : #71c016; color: #71c016\">
                                 별점 및 후기 보기</a>" +
                  "          </li>" +
                  "          <li class=\"nav-item\">" +
                  "            <a class='nav-link' id='question-tab' data-toggle='tab' href='#question'
                                 role=\"tab\" aria-controls=\"question\" aria-selected=\"false\"
                                 style=\"border-bottom-color : #71c016; color: #71c016\">
```

```
                                문의 보기</a>" +
                    "          </li>" +
                    "        </ul>"+
                    "        <div class=\"tab-content py-0 px-0\">" +
//상세정보 시작
                    "          <div class='tab-pane fade active in show' id='detailInfo' role='tabpanel'
                                aria-labelledby=\"detailInfo-tab\">" +
                    "            <div class=\"d-flex flex-wrap justify-content-xl-between\">" +
                    "              <div class=\"d-flex border-md-right flex-grow-1
                                     align-items-center justify-content-center p-3 item\">" +
                    "                <li class=\"nav-item d-none d-lg-block w-100\">" +
                                       dp.get("AD_CONTENT")+
                                       //"광고 작성일 : " + dp.get("AD_DATE").toString()+
                    "                </li>" +
                    "              </div>" +
                    "            </div>" +
                    "          </div>" +
//상세정보끝, 후기시작
                    "          <div class=\"tab-pane fade\" id=\"review\" role=\"tabpanel\"
                                aria-labelledby=\"review-tab\">" +
                    "            <div class=\"d-flex flex-wrap justify-content-xl-between\">"+
                    "              <div class=\"d-flex border-md-right flex-grow-1
                                     align-items-center justify-content-center p-3 item\">"+
                    "                <li class=\"nav-item d-none d-lg-block w-100\">"+
                    "                  <p>총 "+rvList.size()+"건의 후기가 있습니다.</p> 평점:");
//평점 계산
                    if(rvList.size()!=0) {
                     double pSum=0;
                     int oneStarCnt=0;
                     int twoStarCnt=0;
                     int threeStarCnt=0;
                     int fourStarCnt=0;
                     int fiveStarCnt=0;
                     for(int i=0;i<rvList.size();i++) {
                      String ptStr= String.valueOf(rvList.get(i).get("RV_STPOINT"));
                      int pt = Integer.valueOf(ptStr);
                      pSum += pt;
                      if(pt==1) oneStarCnt++;
                      else if(pt==2) twoStarCnt++;
                      else if(pt==3) threeStarCnt++;
                      else if(pt==4) fourStarCnt++;
                      else if(pt==5) fiveStarCnt++;
                     }
                    double pAvg = Math.round((pSum/(double)rvList.size()*10)/10.0);
                    sb.append(pAvg+". 1점인원:"+oneStarCnt+"명. 2점인원:"+twoStarCnt+"명.
                              3점인원:"+threeStarCnt+"명. 4점인원:"+fourStarCnt+"명. 5점인원:"+fiveStarCnt);
                    }
//평점 끝
          sb.append("<div class=\"table-responsive\">"+
                     "<table class=\"table table-hover table-condensed\">"+
                     " <thead>"+
                     "  <tr>"+
                     "   <th>별점</th>"+
                     "   <th>구매옵션</th>"+
                     "   <th>구매일</th>"+
                     "   <th>내용</th>"+
                     "   <th>작성자</th>"+
                     "   <th>작성일</th>"+
                     "  </tr>"+
                     " </thead>"+
                     " <tbody>");
          for(int i=0; i<rvList.size();i++) {
           sb.append(" <tr>"+
                      "  <td>"+String.valueOf(rvList.get(i).get("RV_STPOINT"))"</td>"+
```

```
                    "  <td>");
            if(rvList.get(i).get("OP_CONTENT")!=null)
              sb.append(rvList.get(i).get("OP_CONTENT")+"/"); else sb.append("선택 옵션 없음");
            if(rvList.get(i).get("OP_PERIOD")!=null)
              sb.append("기간:"+rvList.get(i).get("OP_PERIOD")+"/"); else sb.append("");
            if(rvList.get(i).get("OP_TIMES")!=null)
              sb.append(""+String.valueOf(rvList.get(i).get("OP_TIMES"))+"회/"); else sb.append("");
            if(rvList.get(i).get("OP_DAY")!=null)
              sb.append(""+rvList.get(i).get("OP_DAY")+"/"); else sb.append("");
            if(rvList.get(i).get("OP_CLOCK")!=null)
              sb.append("시간:"+rvList.get(i).get("OP_CLOCK")); else sb.append("");
            sb.append("</td>"+
                      "<td>"+String.valueOf(rvList.get(i).get("PS_DATE"))+"</td>"+
                      "<td>");
//후기내용이 길다면 더보기 버튼 넣어주기
            if(rvList.get(i).get("RV_CONTENT").length()>20) {
              sb.append("<p>"+rvList.get(i).get("RV_CONTENT").substring(0, 20)+
                        " <p class='text-success showHiddenReview'>
                           더보기<i class=\"mdi mdi-arrow-down\"></i>
                          </p>"+
                        " <p class='hiddenReview' hidden>"+rvList.get(i).get("RV_CONTENT").substring(20)+
                        " </p>"+
                        " <p class='text-success foldHiddenReview' hidden>
                           접기<i class=\"mdi mdi-arrow-up\"></i>
                          </p>");
// 후기내용이 짧다면 그냥 넣어주기
          }else {
            sb.append("<p>"+rvList.get(i).get("RV_CONTENT")+"</p>");
          }
          sb.append("</td>"+
                    "<td>"+rvList.get(i).get("RV_NAME")+"</td>"+
                    "<td>"+String.valueOf(rvList.get(i).get("RV_DATE"))+"</td>"+
                   "</tr>");
          }//for END
          sb.append("       </tbody>"
                   + "     </table>"
                   + "    </div>"
                   + "   </li>"
                   + "  </div>"
                   + " </div>"
                   + "</div>"+
//후기 끝//문의시작
                   "<div class=\"tab-pane fade\" id=\"question\" role=\"tabpanel\"
                     aria-labelledby=\"question-tab\">"+
          makeHtmlQna(qaList)+
                   "</div>"+//id='question'
//문의 끝
                   "                                  </div>" +
                   "                              </div>" +
                   "                          </div>" +
                   "                      </div>" +
                   "                  </div>" +
                   "               </div>");
          return sb.toString();
          }
```

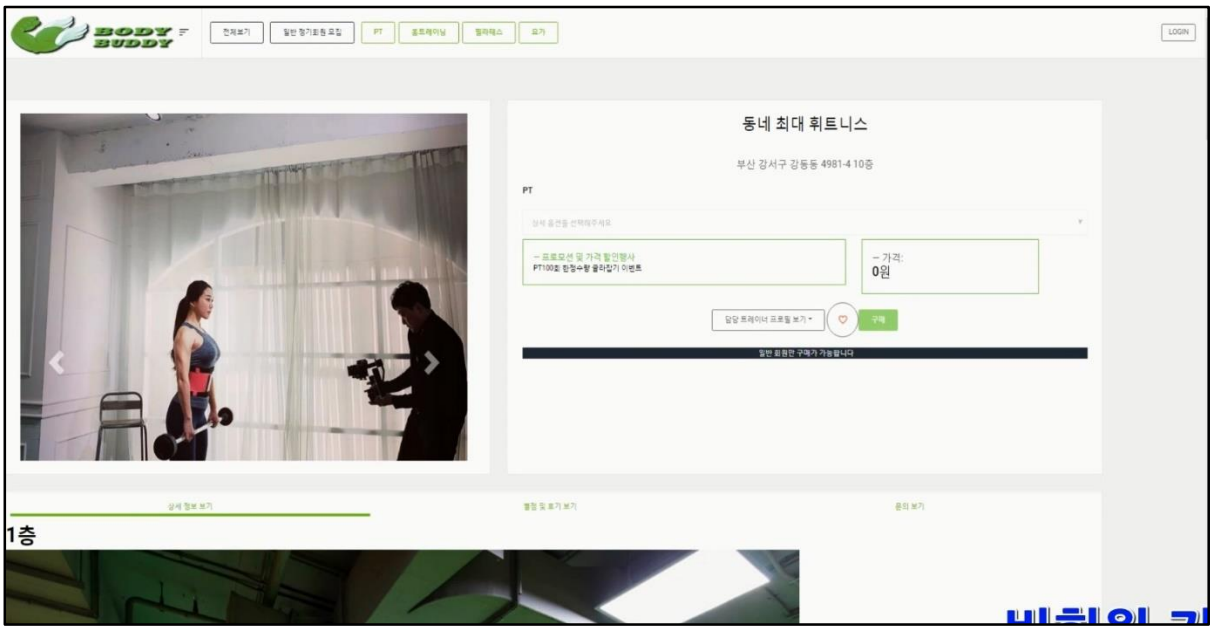
## ■ DAO & MAPPER

```
<select parameterType="String" id="detailPage" resultType="HashMap">
                   SELECT DC.*, AD.AD_CONTENT
                   FROM     (SELECT DISTINCT AD_CODE, AD_NAME, AD_TITLE, AD_STATUS, AD_DATE,AD_CATEGORY,
                                     M_ADDR, DP.T_CID, M_NAME, M_KIND, T_CID_ADDR
                             FROM DETAILPAGE DP
                             LEFT JOIN T_CID_ADDR TCA ON DP.T_CID=TCA.T_CID) DC
                   LEFT JOIN AD ON DC.AD_CODE=AD.AD_CODE
                   WHERE DC.AD_CODE= #{ad_code}
```

```
</select>
<select parameterType="String" id="detailReview" resultType="HashMap">
                SELECT RV_NUM, RV_PSCODE, RV_NAME, RV_CONTENT, RV_DATE, RV_STPOINT,
                AD_CODE, PS_DATE, OP_CODE, OP_CONTENT, OP_PERIOD, OP_TIMES, OP_DAY, OP_CLOCK
                FROM RV
                LEFT JOIN PS ON RV.RV_PSCODE=PS.PS_CODE
                LEFT JOIN AD ON PS.PS_ADCODE=AD.AD_CODE
                LEFT JOIN OP ON PS.PS_OPCODE=OP.OP_CODE
                WHERE AD.AD_CODE = #{ad_code}
</select>
```

## ■ 결과

**-상세 페이지 구성-**




**-옵션 및 가격, 각종 버튼 상세-**

**-트레이너 프로필 보기 DropDown-**

## -별점 및 후기보기-

## -문의보기-

## 4 JOB_NAME : 회원 가입 작성자 : 박한솔

### ■ 개요

1. 회원정보를 입력 받아 저장한다.

### ■ CONTROLLER

```
@RequestMapping(value = "/normalmemberjoin", method = RequestMethod.POST)
        public ModelAndView normalMemberJoin(Member mb) {

                mav = mm.normalMemberJoin(mb);

                return mav;
        }
```

회원정보를 mb 에담아서 넘긴다

### ■ SERVICE

```
        public ModelAndView normalMemberJoin(Member mb) {
                System.out.println("맴버 매니지맨트 시작");
                mav = new ModelAndView();
                // 비번을 암호화(Encoding)할 수 있지만 복호화(Decoding)는 불가능
                BCryptPasswordEncoder pwdEncoder = new BCryptPasswordEncoder();
                mb.setM_pw(pwdEncoder.encode(mb.getM_pw()));
                System.out.println("asdasdsadasd    = " + mb.getM_phone());

                if (mDao.normalMemberJoin(mb)) {
                        view = "loginJoinFrm/loginFrm";
                } else {
                        view = "loginJoinFrm/join";
                }
                mav.setViewName(view);
                System.out.println("매니지맨트 종료");

                return mav;
        }
```

### ■ DAO & MAPPER

```
        public boolean normalMemberJoin(Member mb);

        <insert id="normalMemberJoin" parameterType="member">
                INSERT INTO M

        VALUES(#{m_id},#{m_pw},#{m_name},#{m_phone},TO_DATE(#{m_birth},'YYYY/MM/DD'),#{m_addr},#{m_kind},#{m_exaddr})
        </insert>
```

받아온 회원의 정보들을 테이블에 INSERT 하여 저장한다

### ■ 결과

아이콘이 있으면 필수내용입니다

x5133853@naver.com 중복 확인

JB3HQ8IC68 인증번호확인

••••••

••••••

박한슬

01045689593

19940402

서울 강남구 강남대로 238 주소 검색

102동 102호

모든 회원 약관에 동의합니까?

회원가입

이미 회원가입 하셨나요? 로그인

**정보를 전부 입력후 회원가입 버튼을 누르면 회원가입이 완료가 된다**

## 5 JOB_NAME : 아이디 중복체크 작성자 : 박한솔

### ■ 개요

1. 클라이언트의 아이디를 중복확인

### ■ CONTROLLER

```
@ResponseBody
        @RequestMapping(value = "/checkid", method = RequestMethod.POST)
        public int checkId(String m_id) {
                int checkcon = mm.checkId(m_id);
                return checkcon;
        }
```

Ajax 를 이용 하여 id 값을 가져온후 SERVICE 에 checkId 를 실행

### ■ SERVICE

```
public int checkId(String m_id) {
                int check = 0;
                String s = m_id;
                session.setAttribute("m_id", m_id);
                check = mDao.checkId(m_id);

                return check;
        }
```

가져온 id 값을 세션에 저장후 dao 로 보낸다

### ■ DAO & MAPPER

```
        public int checkId(String s);
-----------------------------------------------------------------
        <select id="checkId" parameterType="String" resultType="Integer">
                SELECT
                COUNT(*) FROM M
                WHERE M_ID = #{m_id}
        </select>
```

가져온 id 값을 SELECT 하여 0 또는 1 을 반환한다

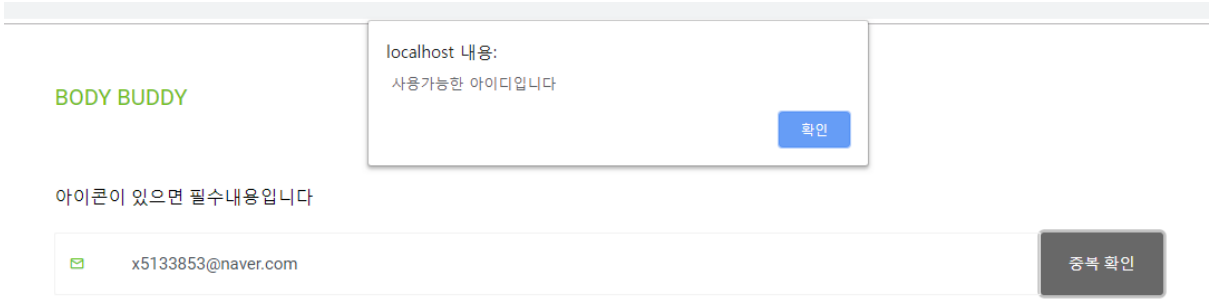
### ■ FRONT-END

```
$('#idCheck').click(function() {
                oldid = $('#m_id').val();
                $.ajax({
                        url : "checkid",
                        type : "post",
                        data : {
                                "m_id" : $('#m_id').val()
                        },
                        dataType : "html",
                        /*data:{m_id : $('#m_id').val(), sdf:"sdfsdfdfsdf"},*/
                        success : function(data) {
                                if (data < 1) {
                                        alert(" 사용가능한 아이디입니다  ");
                                        console.log(data);
                                        console.log(m_id);
                                        $('#sendRndNum').prop("disabled", false);
                                } else {
                                        alert(" 중복된 아이디입니다 ");
                                        $('#joinbtn').prop("disabled", true);

                                }
                        },
                        error : function(error) {
```

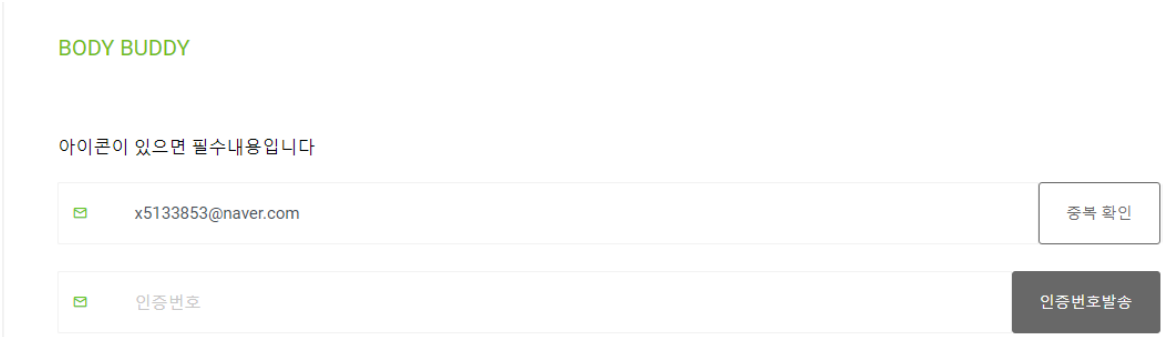
```
                                    console.log(error);
                                    alert(" 실패 ");

                        }
                });

        });
```

DB 에서 가져온값이 1 보다 작으면 alert 으로 사용가능한 아이디표시 및 메일인증 버튼 활성화 그외에는 중복된아이임을 표시

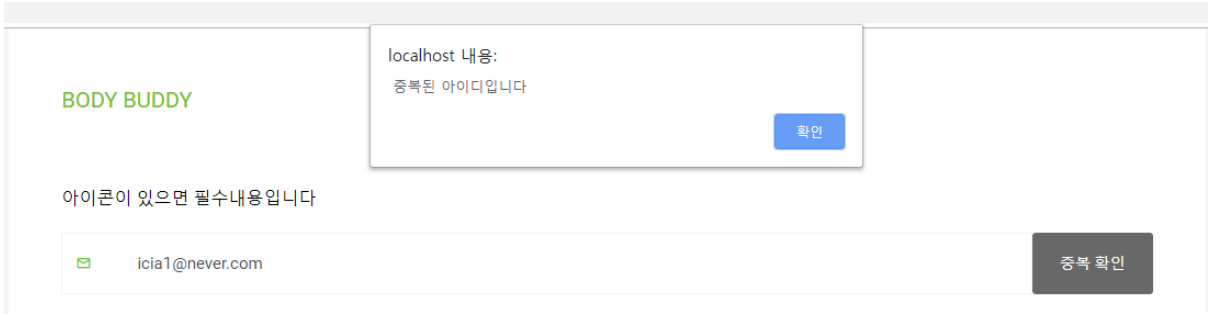
## ■ 결과




**1 아이디입력후 중복확인 클릭**




**2 사용가능한 아이디일경우 인증번호 발송버튼 활성화**



## 1 중복된 아이디일시

아이콘이 있으면 필수내용입니다

icia1@never.com 중복 확인

인증번호 인증번호발송

## 2 인증번호 버튼 비활성화

## 6 JOB_NAME : 이메일 인증 작성자 : 박한솔

### ■ 개요

1. 아이디(이메일)로 메일을 보내 일치할경우 회원가입 버튼 활성화

### ■ CONTROLLER

(코드)

```
@ResponseBody
@RequestMapping(value = "/sendrndnum")
public String sendRndNum() {

        System.out.println("메일 보내기");
        String setfrom = "qhelqkel@gmail.com"; // 보내는 아이디
        String title = "BodyBuddy 인증번호"; // 제목

        String m_id = (String) session.getAttribute("m_id"); // 받는사람아이디
        String certification = mm.getRamdomPassword(10);// 내용( 인증번호 )
        System.out.println(m_id);
        try {
                MimeMessage message = mailSender.createMimeMessage();
                MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-
8");

                messageHelper.setFrom(setfrom); // 보내는사람 ( qhelqkel@gmail.com )
                messageHelper.setTo(m_id); // 받는사람 이메일( 중복확인한 아이디 )
                messageHelper.setSubject(title); // 메일제목
                messageHelper.setText("인증번호 : " + certification); // 메일 내용,인증번호

                mailSender.send(message);
        } catch (Exception e) {
                System.out.println(e);
        }

        return certification;
}
```

(설명)

getRamdomPassword(10)를 실행하고 10 글자의 랜덤문자를 certification 에 저장

### ■ SERVICE

(코드)

```
public String getRamdomPassword(int len) {
        char[] charSet = new char[] { '0', '1', '2', '3', '4', '5',
                        '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F','G',
                        'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R',
                        'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
        int idx = 0;
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < len; i++) {
                idx = (int) (charSet.length * Math.random());
                System.out.println("idx ::::" + idx);
                sb.append(charSet[idx]);
        }
        return sb.toString();
}
```

(설명)

랜덤 함수를 이용하여 배열안의 문자를 가져온 수만큼의 문자를 생성

### ■ FRONT-END

```
var mail = null
        $('#sendRndNum').click(function() {
                $.ajax({
                        url : "sendrndnum",
                        type : "post",
                        dataType : "html",
                        /*data:{m_id : $('#m_id').val(), sdf:"sdfsdfdfsdf"},*/
                        success : function(data) {
                                $('#sendRndNumCheck').prop("type", "button");
                                $('#sendRndNum').prop("type", "hidden");
                                mail = data;
                                console.log("data" + data);
                                console.log(mail);
                                alert("인증번호를  발송하였습니다")
                        },
                        error : function(error) {
                                console.log(error);
                                alert(" 실패 ");

                        }
                });//end ajax

        });//end click
        //인증번호 체크
        $("#sendRndNumCheck").click(function() {
                console.log(mail);
                if ($("#emailNum").val() == mail) {
                        $('#joinbtn').prop("disabled", false);
                        alert("인증되었습니다")
                } else {
                        alert("인증번호를 다시 확인해주세요")
                }
        })
```

반환된 문자를 `mail` 변수에 담아 클라이언트가 입력한 값과 비교하여 같으면 회원가입 버튼을 활성화 같지않으면 다시 확인 해달라는 경고창을 출력

## ■ 결과

아이콘이 있으면 필수내용입니다

x5133853@naver.com 중복 확인

인증번호 인증번호확인

패스워드를 입력해주세요

패스워드 확인

이름을 입력해주세요

전화번호를 입력해주세요 (-)제외

생년월일 8자리를 입력해주세요 ex)19900601

시/도-군/구 주소 검색

상세주소 입력

모든 회원 약관에 동의합니까?

회원가입

**이메일 인증 전에는 회원가입 버튼 비활성화**

시/도-군/구 주소 검색

상세주소 입력

모든 회원 약관에 동의합니까?

회원가입

**메일인증후 회원가입 버튼 활성화**

## 7 JOB_NAME : 로그인 작성자 : 박한솔

■ 개요

1. 클라이언트가 서비스 이용을 위해 로그인을 한다

■ CONTROLLER

```
@RequestMapping(value = "/access", method = RequestMethod.POST)
        public ModelAndView access(Member mb) {
                mav = ks.access(mb);
                return mav;
        }
```

아이디와 비밀번호를 mb 에담아서 넘겨준다

■ SERVICE

```
public ModelAndView access(Member mb) {
                mav = new ModelAndView();
                String view = null;
                String checkId = kDao.deleteRealIdCheck(mb.getM_id());
                if (checkId != null)  {
                        mav.setViewName("loginJoinFrm/loginFrm");
                        mav.addObject("loginCheck", "이미 탈퇴 처리된 계정입니다");
                        return mav;
                }
                BCryptPasswordEncoder pwdEncoder = new BCryptPasswordEncoder();
                String pwdEncode = kDao.getSecurityPwd(mb.getM_id());
                if (pwdEncode != null) {
                        if (pwdEncoder.matches(mb.getM_pw(), pwdEncode)) {
                                String kind = kDao.getMemberKind(mb.getM_id());
                                switch (kind) {
                                case "n":
                                        mb = kDao.getNormalInfo(mb.getM_id());
                                        break;
                                case "t":
                                        mb = kDao.getTrainerInfo(mb.getM_id());
                                        break;
                                case "c":
                                        mb = kDao.getCompanyInfo(mb.getM_id());
                                        break;
                                }
                                session.setAttribute("mb", mb);
                                view = "forward:/";
                        }else {
                                view = "loginJoinFrm/loginFrm";
                                mav.addObject("loginCheck", "다시 입력해주세요(Password Error)");
                        }
                }else {
                        view = "loginJoinFrm/loginFrm";
                        mav.addObject("loginCheck", "다시 입력해주세요(Email Error)");
                }
                mav.setViewName(view);
                return mav;
        }
```

가져온 아이디 값으로 deleteRealIdCheck 을 실행 하여 회원 탈퇴여부를 먼저 확인후 탈퇴일경우 “탈퇴한 계정입니다” 를 출력하며 탈퇴회원이 아닐 경우 DB 테이블을 검색한다. 이때 가져온 PW 를 암호화 하여 비교하고 일치하면 SWITCH 로넘어가며 KIND 가 N 이면 노말회원 T 면트레이너 회원 C 는 업체회원으로 나뉘어진다.

■ DAO & MAPPER

```
        public Member getNormalInfo(String m_id);

        public Member getTrainerInfo(String m_id);
```

```
public Member getCompanyInfo(String m_id);

<select parameterType="String" id="getNormalInfo" resultType="member">
SELECT * FROM NINFO WHERE M_ID = #{m_id}
</select>

<select parameterType="String" id="getTrainerInfo" resultType="member">
SELECT * FROM TINFO WHERE M_ID = #{m_id}
</select>


<select parameterType="String" id="getCompanyInfo" resultType="member">
SELECT * FROM CINFO WHERE M_ID = #{m_id}
</select>
```

미리 조인 하여 만든 뷰를 이용하여 필요한 정보들만 가져온다

로그인 이후 메인페이지로 이동된다.

## 8 JOB_NAME : 아이디 찾기 작성자 : 박한솔

### ■ 개요

1. 클라이언트가 입력했었던 정보중 일부를 입력하여 일치하는 아이디 값을 출력

### ■ CONTROLLER

```
@RequestMapping(value = "/forgetidsearch", method = RequestMethod.POST)
        public ModelAndView forgetIdSearch(Member mb) {
                mav = mm.forgetId(mb);

                return mav;
        }
```

클라이언트가 입력한 정보를 mb 에 담아서 forgetId 에 담아 넘긴다

### ■ SERVICE

```
public ModelAndView forgetId(Member mb) {
                List<Member> tList = null;

                tList = mDao.forgetId(mb);
                mav = new ModelAndView();
                if (tList.size() == 0) {
                        mav.addObject("m_id", "일치하는정보가 없습니다");
                } else {
                        String m_id = "";
                        for (int i = 0; i < tList.size(); i++) {
                                m_id += tList.get(i).getM_id() + "<br>";
                                System.out.println(m_id);
                        }
                        mav.addObject("m_id", m_id);
                        mav.addObject("forget", "id");
                } // else

                view = "loginJoinFrm/new";
                mav.setViewName(view);
                return mav;

        }
```

가져온 값을 DAO 에넘겨 일치하는 정보가 있을시 addObject 에 담아 EL 로 출력할수있게하고 일치하는 정보가없을 경우 일치하는정보가 없습니다를 출력한다.

### ■ DAO & MAPPER

```
public List<Member> forgetId(Member mb);

        <select id="forgetId" parameterType="member" resultType="member">
                SELECT
                M_ID FROM M
                WHERE M_NAME = #{m_name} AND M_PHONE = #{m_phone} AND
                M_BIRTH = #{m_birth}
        </select>
```

가져온 정보로 검색하여 member 에 담는다

### ■ 결과

BODY BUDDY

아이디

박한솔

01045689593

19940402

찾기

비밀번호를 찾으시나요?

BODY BUDDY

아이디를 잊으셨습니까?

찾으시는 아이디는 iciat11@never.com
입니다.

## 9 JOB_NAME : 웹소켓을 활용한 실시간 알림 기능 작성자 : 장기림

### ■ 개요

1. 로그인 시 웹소켓 연결 및 자신에게 온 알림 출력
2. 알림 받아보는 시간에 따라 점차적으로 시간 출력(예: 방금 전, 3 시간 전, 2 일 전)
3. 접속중/ 비접속중에 따라 알림 보내기 여부 결정
4. JSON 형태로 데이터를 받아와서 활용.
5. 구매 및 판매, 문의도착, 문의답변, 소속요청, 소속수락, 소속끊기 이벤트 발생시 알림 보내고 받음.
6. 읽지 않은 알림 개수에 따라 숫자 표시
7. 받은 알림 클릭 시 알림 제거 가능.

### ■ WEB-SOCKET HANDLER

```
public class AlarmHandler extends TextWebSocketHandler {
        @Autowired
        KirimService ks;
        private String json;
        private WebSocketSession rSession;
        private Map<String,String> ws;
        private String rId;
        private static Logger logger = LoggerFactory.getLogger(AlarmHandler.class);
        //접속자 관리
        //private static Set<Session> clients = Collections.synchronizedSet(new HashSet<Session>());
        private static List<WebSocketSession> wSessionList = new ArrayList<WebSocketSession>();
        private static Map<String,WebSocketSession> idAndSession = new HashMap<>();
        @Override// 클라이언트와 연결 이후에 실행되는 메서드 //@OnOpen역할
        public void afterConnectionEstablished(WebSocketSession wSession) throws Exception {
                wSessionList.add(wSession);
        }
        @Override//클라<->서버간 상호작용 메소드 //@OnMessage 역할
        protected void handleTextMessage(WebSocketSession wSession, TextMessage message)
                                        throws Exception {
                String jsonObj = new Gson().toJson(message);
                Map<String,String> m =null;
                String m_id=null;
                logger.info("jsonObj : "+jsonObj);
                logger.info("{}로 부터 {} 받음", wSession.getId(), message.getPayload());
                String sort = message.getPayload().substring(0, 4);
                switch(sort) {
                case "CON1": //접속시 모든 알림메세지 찍어줌
                        m_id = message.getPayload().substring(4);
                        idAndSession.put(m_id, wSession);
                        json = ks.alarmReceiveAll(m_id);
                        wSession.sendMessage(new TextMessage(json.toString()));
                        break;
                case "PS01": //구매 및 판매
                        m = fromJson(message);
                        ks.alarmSelfPurch(m);//구매자가 자기구매알림 받기
                        ws = ks.alarmSendPurch(m);//구매자가 판매자에게 구매알림보내기
                        sendOnline(ws);
                        break;
                case "QA01"://문의 도착
                        m = fromJson(message);
                        ws = ks.alarmSendQuestion(m);//문의자가 답변자에게 문의알림보내기
                        sendOnline(ws);
                        break;
                case "AQ01"://문의 답변
                        m = fromJson(message);
                        ws = ks.alarmSendAnswer(m);//답변자가 문의자에게 답변알림보내기
```

```
                        sendOnline(ws);
                        break;
                case "SS01"://소속 요청
                        m = fromJson(message);
                        ws = ks.alarmSendSskReq(m);//소속요청 트레이너가 업체에게 요청알림보내기
                        sendOnline(ws);
                        break;
                case "SS02"://소속 수락
                        m = fromJson(message);
                        ws = ks.alarmSendSskAcpt(m);//업체가 요청 트레이너에게 수락알림보내기
                        sendOnline(ws);
                        break;
                case "SS03"://소속 거절
                        m = fromJson(message);
                        ws = ks.alarmSendSskRjct(m);//업체가 요청 트레이너에게 거절알림보내기
                        sendOnline(ws);
                        break;
                case "SS04"://소속 끊기
                        m = fromJson(message);
                        ws = ks.alarmSendSskCut(m);//업체가 트레이너에게 소속끊기알림보내기
                        sendOnline(ws);
                        break;

                }//SWITCH END
        }
        private Map<String, String> fromJson(TextMessage message) {//JSON에서 HashMap으로 변환
                json = message.getPayload().substring(4);
                return new Gson().fromJson(json,new TypeToken<Map<String,String>>(){}.getType());
        }
        private void sendOnline(Map<String,String> ws) throws IOException {
                rId = ws.get("al_mrid");
                rSession = idAndSession.get(rId);
                if(rSession!=null) {
                        json = ks.alarmReceiveAll(rId);
                        rSession.sendMessage(new TextMessage(json.toString()));
                }
        }
        @Override // 클라이언트와 연결을 끊었을 때 실행되는 메소드 //@OnClose역할
        public void afterConnectionClosed(WebSocketSession wSession, CloseStatus status)
                                        throws Exception {
                wSessionList.remove(wSession);
        }
}
```

■ SERVICE

```
public Map<String,String> alarmSendPurch(Map<String, String> m) {
                int al_code=0;
                String ad_name=kDao.getAdName(m.get("ad_code"));
                m.put("al_msid", m.get("m_id"));
                m.put("al_mrid", ad_name);
                al_code = kDao.alarmSendPurch(m);
                if(al_code!=0) {
                        m.put("al_code", String.valueOf(al_code));
                        return m;
                }
                return null;
        }
public String alarmReceiveAll(String m_id) {
                String json=null;
                List<Alarm> msgList=null;
                msgList = kDao.alarmReceiveAll(m_id);
                if(msgList!=null) {
                        String html = makeHTMLAllMsg(msgList);
                        List<String> list = new ArrayList<>();
                        list.add(html);
```

```
                        list.add(String.valueOf(msgList.size()));
                        json = new Gson().toJson(list);
                }
                return json;
        }
private String makeHTMLAllMsg(List<Alarm> msgList) {
                StringBuilder sb = new StringBuilder();
                for(int i=0; i<msgList.size(); i++) {
                        String time = strTimeCalc(msgList.get(i).getAl_date());
                        sb.append("        <a class='dropdown-item alarm-confirm'>"
                                        + "                        <input type='hidden'
class='al_code' value='"+msgList.get(i).getAl_code()+"'/>"
                                        + "            <div class='item-content'>"
                                        + "                <h6 class='font-weight-normal'>");
                        if(msgList.get(i).getAl_kind().equals("PS01")) {
                                sb.append(msgList.get(i).getAl_mSId()+"님이 당신의 상품을
구매하였습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("PS02")) {
                                sb.append("상품을 성공적으로 구매하였습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("QA01")) {
                                sb.append("답변을 기다리고있는  문의가 도착했습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("AQ01")) {
                                sb.append("문의에 대한 답변이 도착하였습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("SS01")) {
                                sb.append("트레이너의 소속계약 요청이 들어왔습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("SS02")) {
                                sb.append("소속계약 요청이 수락되었습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("SS03")) {
                                sb.append("소속계약 요청이 거부되었습니다.");
                        }else if(msgList.get(i).getAl_kind().equals("SS04")) {
                                sb.append("업체와의 소속계약이 종료되었습니다.");
                        }
                        sb.append("                                        </h6>"
                                        + "                <p class='font-weight-light small-text
mb-0 text-muted'>"+time+"</p>"
                                        + "            </div>"
                                        + "        </a>");
                }
                return sb.toString();
        }
private String strTimeCalc(LocalDateTime sendTime) {
                String str=null;
                LocalDateTime today = LocalDateTime.now();
                long sec = ChronoUnit.SECONDS.between(sendTime, today);
                long min = ChronoUnit.MINUTES.between(sendTime, today);
                long hour = ChronoUnit.HOURS.between(sendTime, today);
                long days = ChronoUnit.DAYS.between(sendTime, today);
                long weeks = ChronoUnit.WEEKS.between(sendTime, today);
                long months = ChronoUnit.MONTHS.between(sendTime, today);
                long years = ChronoUnit.YEARS.between(sendTime, today);
                if(sec<60) str = "방금 전";
                else if(min<60) str = min+" 분 전";
                else if(hour<24) str = hour+" 시간 전";
                else if(days<7) str = days+" 일 전";
                else if(weeks<5) str = weeks+" 주 전";
                else if(months<12) str = months+" 개월 전";
                else if(years<1000) str = years+" 년 전";
                return str;
        }
```

## ■ DAO & MAPPER

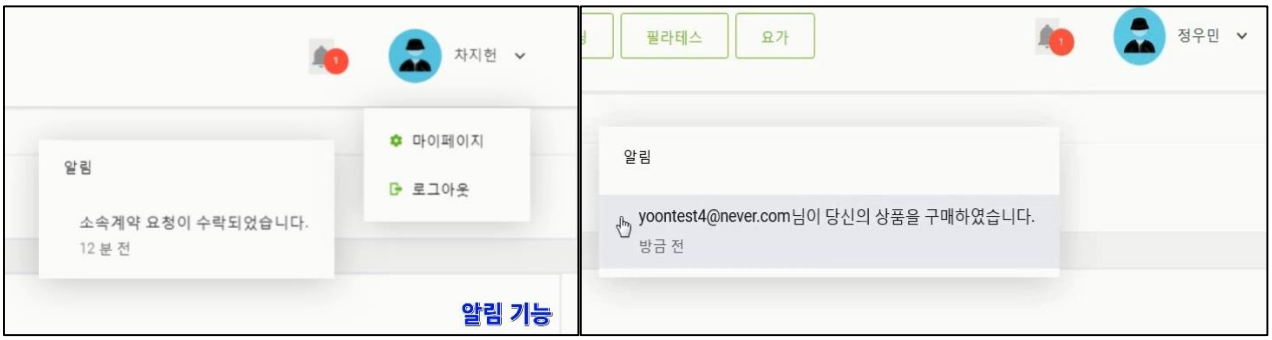
```
public int alarmSendPurch(Map<String, String> m);
```

```
<insert id="alarmSendPurch" parameterType="HashMap">
        INSERT INTO AL VALUES(AL_SEQ.NEXTVAL, #{al_msid},#{al_mrid},'PS01','n',SYSDATE)
        <selectKey keyProperty="al_code" resultType="int" order="AFTER">
                SELECT AL_SEQ.CURRVAL FROM DUAL
        </selectKey>
</insert>

public List<Alarm> alarmReceiveAll(String m_id);
<select id="alarmReceiveAll" parameterType="String" resultType="alarm">
                SELECT * FROM AL WHERE AL_MRID=#{m_id} AND AL_STATUS='n'
 </select>
(Mybatis의 SelectKey를 활용해서 Primary Key를 가져온다.)
```

## ■ FRONT-END

```
var sock=null;
function connect(){
        var connect = new SockJS("/alarm");
        sock=connect;
        $("#msg").val("CON1"+m_id);
}
function onMessage(msg) {
        var data = msg.data;
        console.log("받은메세지...",data);
        var parseData = JSON.parse(data);
        $(".beforeAlarm").after(parseData[0]);
        if(parseData[1]>0){
                $(".count").html("<p class='text-white'+
                                style='font-size: 5px;line-height: 15px;'>"+
                                parseData[1]+"</p>");
                $(".count").prop("hidden", false);
        }
}
window.onload=function(){
        if(kind == 'n' || kind == 't' || kind == 'c'){
                console.log("보내는메세지...",$("#msg").val());
                sock.send($("#msg").val());
                sock.onmessage = onMessage;
                //알림클릭시 본 알림으로 업데이트
                $("#notificationDropdown").click(function(){
                        var ac = $(this).next().children().filter($(".alarm-confirm"));
                        ac.click(function(){
                                var al_code = $(this).children().filter($(".al_code")).val();
                                $.ajax({
                                        url:'alarmconfirm',
                                        type:'post',
                                        data:{al_code:al_code},
                                        dataType:'text',
                                        success:function(suc){
                                                console.log(suc);
                                                ac.remove();
                                        },
                                        error:function(err){
                                                console.log(err);
                                        }
                                });//$.ajax({
                        });//ac.click(function(){
                });//$("#notificationDropdown").click(function(){
        }//if(kind == 'n' || kind == 't' || kind == 'c'){
}//window.onload=function(){
```

## ■ 결과


차지헌
마이페이지
로그아웃
알림
소속계약 요청이 수락되었습니다.
12 분 전
알림 기능
필라테스
요가
정우민
알림
yoontest4@never.com님이 당신의 상품을 구매하였습니다.
방금 전

-알림 구현 예시-

## 10 JOB_NAME : 찜하기 (비회원/일반회원) 작성자 : 장기림

### ■ 개요

1. 세션 맵에 광고 PrimaryKey 를 접붙여서 비회원 찜하기 구현
2. 기본적인 Ajax 비동기통신으로 MVC 흐름에 따라 일반회원 찜하기 구현
3. 버튼의 HTML 을 반환하여 사용
4. 메인 화면 및 상세페이지에서 모두 사용 가능

### ■ CONTROLLER

```
@RequestMapping(value = "/dibsadd", method = RequestMethod.GET)
        @ResponseBody
        public String dibsAdd(@RequestParam("d_adcode") String d_adcode) {
                System.out.println("url:/dibsadd");
                html = ks.dibsAdd(d_adcode);
                return html;
        }
```

### ■ SERVICE

```
public String dibsAdd(String d_adcode) {
                String del = "<button id='dibsDelete" + d_adcode +
                        "'type='button' class='btn btn-outline-danger btn-rounded btn-icon'>"
                                + "<i class='mdi mdi-heart'></i></button>";
                String add = "<button id='dibsAdd" + d_adcode+
                  "'type='button' class='btn btn-outline-secondary btn-rounded btn-icon'>"
                        + "<i class='mdi mdi-heart-outline text-danger'></i></button>";
                String html = null;
                Member sessionMb = (Member) session.getAttribute("mb");
                if (sessionMb == null) {// 비회원 찜 추가
                        session.setAttribute("tempDibs" + d_adcode, "dibs");
                        html = del;
                }else if (sessionMb != null) {// 회원 찜 추가
                        String sessionId = sessionMb.getM_id();
                        Map<String, String> dibs = new HashMap<>();
                        dibs.put("d_adcode", d_adcode);
                        dibs.put("d_id", sessionId);
                        if (kDao.dibsAdd(dibs)) {
                                html = del;
                                return html;
                        }
                        html = add;
                }
                return html;
}
```

### ■ DAO & MAPPER

```
public boolean dibsAdd(Map<String, String> dibs);
<insert id="dibsAdd" parameterType="HashMap">
                INSERT INTO D VALUES(#{d_adcode},#{d_id})
</insert>
```

### ■ FRONT-END

```
function dibsAdd() {
        $(".btn.btn-outline-secondary.btn-rounded.btn-icon").click(function() {
                var dibsAddBtn = $(this);
```

```
                var ad_code = $(this).prop("id").replace("dibsAdd", "");
                $.ajax({
                        type : 'get',
                        url : 'dibsadd',
                        data : {d_adcode : ad_code},
                        dataType : 'html',
                        success : function(data) {
                                console.log(data);
                                console.log(dibsAddBtn);
                                dibsAddBtn.prop("outerHTML", data);
                                dibsDelete();
                        },
                        error : function(err) {
                                console.log(err);
                        }
                }); //ajax End
        });//click secondary
}//end dibsAdd
```

■ 결과

▲상세페이지의 찜하기 버튼

◀메인페이지의 찜하기 버튼

## 11 JOB_NAME : 찜 취소하기 (비회원/일반회원) 작성자 : 장기림

### ■ 개요

1. 세션 맵에 광고 PrimaryKey 를 접붙여서 비회원 찜 취소 구현
2. 기본적인 Ajax 비동기통신으로 MVC 흐름에 따라 일반회원 찜 취소 구현
3. 버튼의 HTML 을 반환하여 사용
4. 메인 화면 및 상세페이지에서 모두 사용 가능

### ■ CONTROLLER

```
@RequestMapping(value = "/dibsdelete", method = RequestMethod.GET)
        @ResponseBody
        public String dibsDelete(String d_adcode) {
                System.out.println("url:/dibsdelete");
                html = ks.dibsDelete(d_adcode);
                return html;
        }
```

### ■ SERVICE

```
public String dibsDelete(String d_adcode) {
                String del = "<button id='dibsDelete" + d_adcode + "' type='button'
                                class='btn btn-outline-danger btn-rounded btn-icon'>"+
                                "<i class='mdi mdi-heart'></i></button>";
                String add = "<button id='dibsAdd" + d_adcode+ "'type='button'
                                class='btn btn-outline-secondary btn-rounded btn-icon'>"+
                                "<i class='mdi mdi-heart-outline text-
danger'></i></button>";
                String html = null;
                Member sessionMb = (Member) session.getAttribute("mb");
                if (sessionMb == null) {// 비회원 찜 제거
                        session.setAttribute("tempDibs" + d_adcode, null);
                        html = add;
                } else if (sessionMb != null) {// 회원 찜 제거
                        String sessionId = sessionMb.getM_id();
                        Map<String, String> dibs = new HashMap<>();
                        dibs.put("d_adcode", d_adcode);
                        dibs.put("d_id", sessionId);
                        if (kDao.dibsDelete(dibs)) {
                                html = add;
                                return html;
                        }
                        html = del;
                }
                return html;
        }
```

### ■ DAO & MAPPER

```
public boolean dibsDelete(Map<String, String> dibs);
<delete id="dibsDelete" parameterType="HashMap">
                DELETE FROM D WHERE D_ADCODE=#{d_adcode} AND D_ID=#{d_id}
</delete>
```

### ■ FRONT-END

```
function dibsDelete() {
        $(".btn.btn-outline-danger.btn-rounded.btn-icon").click(function() {
                console.log($(this));
                var dibsDeleteBtn = $(this);
                var ad_code = $(this).prop("id").replace("dibsDelete", "");
```

```
                $.ajax({
                        type : 'get',
                        url : 'dibsdelete',
                        data : {d_adcode : ad_code},
                        dataType : 'html',
                        success : function(data) {
                                console.log(data);
                                console.log(dibsDeleteBtn);
                                dibsDeleteBtn.prop("outerHTML", data);
                                dibsAdd();
                        },
                        error : function(err) {
                                console.log(err);
                        }
                }); //ajax End
        });//click danger
}//end dibsDelete
```

■ 결과

▲상세페이지의 찜취소 버튼

◀메인페이지의 찜취소 버튼

## 12 JOB_NAME : 구매하기, 옵션 자동 만료 기능(정원) 작성자 : 장기림, 윤상기

### ■ 개요

1. 일반 소비자회원만 구매 가능
2. 다양한 유효성 검사와 적합한 구매조건 위반시 마우스 툴팁이나 버튼제한 등으로 기능제한
3. 구매 전 후로 정원초과 검사하여 광고 및 옵션 만료여부 결정

### ■ CONTROLLER

```
@RequestMapping(value = "/purchsingle", method = RequestMethod.POST)
@ResponseBody
public String purchSingle(Payment ph) {
                System.out.println("URL : /purchsingle");
                html = ks.purchSingle(ph);
                return html;
}
```

### ■ SERVICE

로그인/회원종류 검사 ⇨ 중복구매검사 ⇨ 정원초과검사 ⇨ 구매 ➡ 출결조건 저장 ➡ 옵션만료 확인

```
public String purchSingle(Payment ph) {
                String text = null;
                Member sessionMb = (Member) session.getAttribute("mb");
                if(sessionMb == null) {//로그인 2차검사
                        text = "login";
                        return text;
                }
                Map<String,String> cs = new HashMap<>();//구매정보 저장
                String ps_mid=sessionMb.getM_id();
                String ps_adcode=ph.getPs_adcode();
                cs.put("ps_mid", ps_mid);
                cs.put("ps_adcode", ps_adcode);
                int i=kDao.selectOverlap(cs); //중복구매 검사
                if (sessionMb != null && sessionMb.getM_kind().equals("n") && i==0) {
                        if(opExpirePersonnel(ph)) {//구매 전 정원초과 검사
                                return "full";
                        }
                        ph.setPs_mid(sessionMb.getM_id());
                        kDao.purchSingle(ph); //실제 구매, 구매내역 저장
                        String da_opperiod=ph.getPs_opcode();//실 구매정보 취합
                        String ps_date=ph.getPs_date();
                        Payment ph1=kDao.selectPscode(cs);
                        String ph2=kDao.selectPeriod(da_opperiod);
                        Map<String,String> cs2 = new HashMap<>();
                        String ps_code=ph1.getPs_code();
                        cs2.put("da_period", ph2);
                        cs2.put("ps_code", ps_code);
                        int ps_day=kDao.getPscode(ps_code); //프로그램 시작날짜
                        if(kDao.getCategory(ps_code)!=0) {//일반 프로그램 회원모집인 경우
                                int opperiod2=kDao.selectOpPeriod(ps_code,ph2); {//날짜 계산
                                int endDay = kDao.getEndDay(cs2.get("ps_code"),opperiod2);
                                String str = Integer.toString(ps_day) +
                                                "~" + Integer.toString(endDay);
                                cs2.put("str", str);
```

```
                                kDao.norInsertDaliy(cs2); //구매한 일반 프로그램 출결조건 생성
                        }else if(kDao.getCategory(ps_code)==0){
                         kDao.insertDaliy(cs2); //구매한 트레이닝 프로그램 출결조건 생성
                        }
                        opExpirePersonnel(ph); //구매 후 정원초과 검사
                        text = "success";
                        return text;
                        //트레이너/업체회원 구매제한
                        }else if(sessionMb.getM_kind().equals("c") ||
                                sessionMb.getM_kind().equals("t")){
                                text ="notn";
                                return text;
                }else if(i!=0) {//중복구매제한
                                text ="overlap";
                                return text;
                }
                return text;
        }

private boolean opExpirePersonnel(Payment ph) { //구매 전/후로 정원초과 검사
                Map<String,String> cntper=null;
                cntper = kDao.personnelCalc(ph.getPs_opcode());
                if(cntper!=null) {
                        int cnt = Integer.valueOf(String.valueOf(
                                cntper.get("PURCHCOUNT"))).intValue();
                        int limit = Integer.valueOf(String.valueOf(
                                cntper.get("OP_PERSONNEL"))).intValue();
                        if(limit!=-100) {//모집제한이 있다면
                                if(cnt >= limit){ //정원 초과될 시 옵션 만료 및 광고 만료확인
                                        int result = kDao.expireOption(ph.getPs_opcode());
                                        adExpire();
                                        return true;
                                }
                        }
                }
                return false;
        }
```

■ FRONT-END

```
$("#purchase").click(function(){//구매 버튼 클릭시
        var ad_code = $("#ad_code").val();
        var op_code = $("#optionSelect").prop("selectedOptions")[0].id.replace("op","");
        var $op_code = "#"+op_code;
        var $op_priceValue = $($op_code).val();
        var m_id = "${mb.m_id}";
        console.log(ad_code);
        console.log(op_code);
        console.log($op_priceValue); //옵션 선택에 따라 다르게 출력되는 가격
        $.ajax({
                type:'post',
                url:'purchsingle',
                data: {ps_adcode:ad_code, ps_opcode:op_code, ps_price:$op_priceValue},
                dataType:'text',
                success:function(data){
                        console.log(data);
```

```
                        if(data=='success'){ //구매 성공시
                                //웹소켓 구매알림 보내기
                                var jsonSendPurAlarm = JSON.stringify(
                                                        {ad_code:ad_code,m_id:m_id});
                                $("#msg").val("PS01"+jsonSendPurAlarm);
                                console.log("WebSock3", sock);
                                console.log($("#msg").val());
                                sock.send($("#msg").val());
                                //
                                var sucConf = confirm(
                                "구매에 성공하셨습니다. 내 프로그램 보기로 이동하시겠습니까?");
                                if(sucConf == true){
                                        location.href="infoprogramn?m_id=${mb.m_id}";
                                }else if(sucConf == false){
                                        location.href="#";
                                }
                        }
                        if(data=='full'){
                                alert("해당 상품의 정원이 가득찼습니다.");
                        }
                        if(data=='failed'){
                                alert("구매에 실패하였습니다. 다시 시도해주세요.");
                        }
                        if(data=='login'){
                                var logConf = confirm(
                             "로그인을 해야 구매가 가능합니다. 로그인창으로 이동하시겠습니까?");
                                if(logConf == true){
                                        location.href="login";
                                }else if(logConf == false){
                                        location.href="#";
                                }
                        }
                        if(data=='notn'){
                                alert("일반회원만 결제 가능합니다.")
                        }
                        if(data=='overlap'){
                                alert("이미 구매한 상품입니다.");
                        }
                },
                error:function(err){
                        console.log(err);
                }
        }); //ajax End
});//click End
```

## 13 JOB_NAME : 옵션과 광고 자동 만료 기능(기간) 작성자 : 장기림

### ■ 개요

1. 만료된 광고는 메인리스트에서 제외되어 별도로 관리된다.
2. 한 광고의 모든 옵션이 만료되면 광고까지 만료시킨다.
3. 옵션의 만료 조건
   - 구매할 수 없는 기간일 때 : 특정 시간대(자정)에 메소드를 실행하여 자동 만료기능 구현
   - 정원이 초과했을 때
   - (사용자가 광고를 게시 취소했을 때)

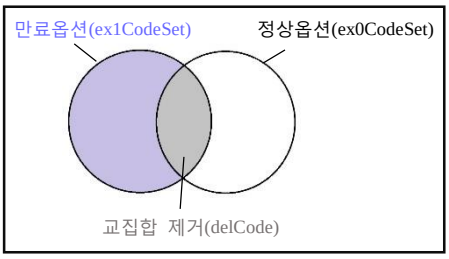
### ■ CONTROLLER

```
@Scheduled(cron="0 0 0 * * * ")
public void adExpirePeriod() {
        System.out.println("스케쥴러 : adExpirePeriod ");
        ks.adExpirePeriod();
}
```

(기간제 만료 조건 : 자정마다 작동하는 메소드 설정)

### ■ SERVICE

```
public void adExpirePeriod() {
        String period=null;
        String periodStartStr=null;
        LocalDate periodStart=null;
        String op_code;
        int result=0;
        List<Map<String,String>> opl = kDao.getOpPeriodList();//옵션 기간(OP_PERIOD) 얻어오기
        LocalDate today = LocalDate.now();//현재 날짜 얻어오기
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyyMMdd");

        for(int i=0;i<opl.size();i++) {
                if(opl.get(i).get("OP_PERIOD").toString().contains("~")) {
                        period = opl.get(i).get("OP_PERIOD").toString();
                        periodStartStr =period.split("~")[0];//시작날짜 문자열
                        periodStart = LocalDate.parse(periodStartStr, formatter);
        //해당 옵션레코드의 시작날짜가 오늘보다 많다면(after라면)
                        if(periodStart.isBefore(today)) { //isBefore:a<b   isAfter:a>b
                                op_code = opl.get(i).get("OP_CODE").toString();
                                result = kDao.expireOption(op_code);
                        }
                }
        }
        adExpire();//해당 광고의 모든 옵션이 만료되었다면 광고를 만료
}
private void adExpire() {
        int result=0;
        String ad_code;
        String ad_code0; // 0:유효옵션
        String ad_code1; // 1:만료옵션
        List<Map<String,String>> aeList =kDao.getAdExpireList();
        Set<String> ex1CodeSet = new HashSet<>();
        Set<String> ex0CodeSet = new HashSet<>();
        for(int i=0;i<aeList.size();i++) {
                if(String.valueOf(aeList.get(i).get("OP_EXPIRE")).equals("1")) {
                        ex1CodeSet.add(aeList.get(i).get("AD_CODE"));
                }else if(String.valueOf(aeList.get(i).get("OP_EXPIRE")).equals("0")) {
                        ex0CodeSet.add((String) aeList.get(i).get("AD_CODE"));
                }
        }
```




```
//1만료HashSet에는 있고 0정상HashSet셋에는 없으면 만료시킴
                System.out.println("ex0CodeSet========"+ex0CodeSet);
                System.out.println("ex1CodeSet========"+ex1CodeSet);
                List<String> delCode = new ArrayList<>();
                Iterator<String> itr0 = ex0CodeSet.iterator();
                while(itr0.hasNext()) {
                        ad_code0 = itr0.next();
                        Iterator<String> itr1 = ex1CodeSet.iterator();
                        while(itr1.hasNext()) {
                                ad_code1 = itr1.next();
                                if(ad_code0.equals(ad_code1)) {//제거할 교집합리스트에 추가
                                        delCode.add(ad_code1);
                                 }
                        }
                }
                for(int i=0;i<delCode.size();i++) {
                        ex1CodeSet.remove(delCode.get(i).toString());
                }
                Iterator<String> comp = ex1CodeSet.iterator();
                while(comp.hasNext()) {
                        ad_code = comp.next();
                        System.out.println("만료시킬 광고 코드 : "+ad_code);
                        result = kDao.expireAd(ad_code);
                        if(result!=0) {
                                System.out.println("만료된광고"+result+"건");
                        }else {
                                System.out.println("만료시킬 광고 없음");
                        }
                }
                System.out.println("정상완료");
        }
```

## ■ DAO & MAPPER

```
<select id="getOpPeriodList" resultType="HashMap">
                SELECT OP_CODE, OP_ADCODE, OP_PERIOD FROM OP
</select>

<update id="expireOption" parameterType="String">
                UPDATE OP SET OP_EXPIRE=1 WHERE OP_CODE=#{op_code}
</update>

<select id="getAdExpireList" resultType="HashMap">
                SELECT AD_CODE, OP_EXPIRE FROM AD
                JOIN OP ON AD_CODE=OP_ADCODE
</select>

<update id="expireAd" parameterType="String">
        <![CDATA[
                UPDATE AD SET AD_STATUS='만료됨' WHERE AD_CODE=#{ad_code}
        ]]>
</update>
```

| 14 | JOB_NAME : 광고 글 쓰기 | 작성자 : 장기림, 고정근 |
|---|---|---|

■ 개요

1. 폼에 입력된 (1)광고 정보, (2)옵션 정보, (3)광고 사진, (4)CKEditor 의 내용을 Multipart 형식으로 Request
2. 트레이너/업체, 일반 정기회원 모집/프로그램 등 조건에 따라서 유효성 검사

■ CONTROLLER

```
@RequestMapping(value = "/adinsert", method = RequestMethod.POST)
        public ModelAndView adinsert(MultipartHttpServletRequest multi) {
                System.out.println("url: /adinsert");
                mav=js.adinsert(multi);
                return mav;
        }
```

■ SERVICE

```
@Transactional
public ModelAndView adinsert(MultipartHttpServletRequest multi) {
                mav = new ModelAndView();
                Question adadd = new Question();
            //데이터 유효 검증을 위해 null로 초기화
                String ad_category = null;
                String ad_title = null;
                String ad_content = null;
                String[] op_contentValues = null;
                String[] day1 = null;
                List<String> ad_nameTId = null;
                String[] op_trainerValues = null;
                String[] dayValues = null;
                String[] op_periodValues = null;
                String[] op_period_1sValues = null;
                String[] op_period_2sValues = null;
                String[] op_timesValues = null;
                String[] op_clock1Values = null;
                String[] op_clock2Values = null;
                String[] op_priceValues = null;
                String[] op_personnelValues = null;
                Member mb = (Member) session.getAttribute("mb");
                adadd.setAd_name(mb.getM_id());
                //광고 카테고리
                ad_category = multi.getParameter("ad_category");
                if (ad_category != null && ad_category != "") {
                        System.out.println("ad_category : " + ad_category);
                        if (ad_category.equals("normal"))
                                adadd.setAd_category("일반");
                        if (ad_category.equals("pt"))
                                adadd.setAd_category("PT");
                        if (ad_category.equals("homeTraining"))
                                adadd.setAd_category("홈트레이닝");
                        if (ad_category.equals("pilates"))
                                adadd.setAd_category("필라테스");
                        if (ad_category.equals("yoga"))
                                adadd.setAd_category("요가");
                }
                //광고 제목
```

```
ad_title = multi.getParameter("ad_title");
if (ad_title != null && ad_title != "") {
        System.out.println("ad_title : " + ad_title);
        adadd.setAd_title(ad_title);
}
//광고 안내 상세 내용(CKEditor)
ad_content = multi.getParameter("ad_content");
if (ad_content != null && ad_content != "") {
        System.out.println("ad_content : " + ad_content);
        adadd.setAd_content(ad_content);
}
op_contentValues = multi.getParameterValues("op_content");//옵션명
op_timesValues = multi.getParameterValues("op_times");//횟수
op_clock1Values = multi.getParameterValues("op_clock1");//시작시간
op_clock2Values = multi.getParameterValues("op_clock2");//끝나는시간
op_priceValues = multi.getParameterValues("op_price");//가격
op_personnelValues = multi.getParameterValues("op_personnel");//정원
op_trainerValues = multi.getParameterValues("op_trainer");


//담당 트레이너
if(mb.getM_kind().equals("t")){
        ad_nameTId = new ArrayList<>();
        for(int i=0;i<op_trainerValues.length; i++) {
                if(op_trainerValues[i]!="") {
                    String[] ad_nameT = op_trainerValues[i].split(",");
                     String ad_nameTName = ad_nameT[0];
                }
        }
}
//요일
dayValues = multi.getParameterValues("day");
String b = "";
for (int i = 0; i < dayValues.length; i++) {
        if (dayValues[i] != "") {
                b += dayValues[i];
        }
}
day1 = b.split(" ");
//일반 일 수
op_periodValues = multi.getParameterValues("op_period");
//기간 시작
op_period_1sValues = multi.getParameterValues("op_period1");
//기간 끝
op_period_2sValues = multi.getParameterValues("op_period2");
adadd.setAd_status("모집중");
Map<String, String> opMap = new HashMap<String, String>();
String xxx;
if (mDao.adinsert(adadd)) { //광고 등록
        xxx = adadd.getXxx();
       //광고 사진 저장
        String root = multi.getSession().getServletContext().getRealPath("/");
        System.out.println("root=" + root);
        String path = root + "resources/upload/";
        File dir = new File(path);
        if (!dir.isDirectory()) {
```

```
                dir.mkdirs();
            }
            Map<String, String> map = new HashMap<>();
            List<MultipartFile> files = multi.getFiles("ap_image");
            String oriFileName = null;
            String sysFileName = null;
            for (int i = 0; i < files.size(); i++) {
              oriFileName = files.get(i).getOriginalFilename();
              String index = String.valueOf(i);
              sysFileName = i+System.currentTimeMillis() + "."
                    +oriFileName.substring(oriFileName.lastIndexOf(".") + 1);
              map.put("ap_adcode", xxx);
              map.put("ap_image", sysFileName);
             if (sysFileName != null) {
               files.get(i).transferTo(new File(path + sysFileName));
                 if (mDao.adPhotoInsert(map)) {//광고 사진 등록
                    System.out.println(i + "번째 광고사진 업로드 및 등록 성공");
                 }
              }
              for (int i = 0; i < op_contentValues.length; i++) {
                    opMap.put("op_adcode", xxx);
                     opMap.put("op_content", op_contentValues[i]);
                     opMap.put("op_price", op_priceValues[i]);
                     if(mb.getM_kind().equals("t")){
                       if(ad_nameTId.size()!=0)
                         opMap.put("op_trainer", ad_nameTId.get(i));
                     }else if(mb.getM_kind().equals("c")) {
                       if (op_trainerValues[i] != "")
                         opMap.put("op_trainer", op_trainerValues[i]);
                     }
                    if(ad_category.equals("normal")) {//일반일때
                      opMap.put("op_period", op_periodValues[i]);
                      opMap.put("op_personnel", "-100" );
                      opMap.put("op_times", op_timesValues[i]);
                      opMap.put("op_day", day1[i]);
                    }else {//일반 제외 다른 프로그램일때
                      if (op_period_1sValues[i] != "" && op_period_2sValues[i] != "") {
                        opMap.put("op_period", op_period_1sValues[i]
                          + "~" + op_period_2sValues[i]);
                      }
                      if (op_timesValues[i] != "") {
                        opMap.put("op_times", op_timesValues[i]);
                      }
                      if (day1 != null && day1.length != 0 && day1[i] != "") {
                        opMap.put("op_day", day1[i]);
                      }
                      if (op_personnelValues[i] != "") {
                        opMap.put("op_personnel", op_personnelValues[i]);
                      }
                    }
                    if (op_clock1Values[i] != "" && op_clock2Values[i] != "") {
                       opMap.put("op_clock", op_clock1Values[i]
                         + "~" + op_clock2Values[i]);
                     }
                     if (mDao.opinsert(opMap)) {{//옵션 입력
                            System.out.println(i + "번째 옵션입력성공");
                     }
              }{//옵션 입력 반복문 끝
        } else {
            System.out.println("광고등록실패");
```

```
                }
                if(mb.getM_kind().equals("t")){
                        mav.setViewName("forward:/trainer");
                }else if(mb.getM_kind().equals("c")) {
                        mav.setViewName("forward:/");
                }
                return mav;
        }
```

1. **일반 정기 회원 모집 등록/ 프로그램 등록에 따라 분류, 데이터 유효성 if 문으로 검사**
2. 입력 받은 값을 배열 변수에 저장 할 수 있도록 선언한다.
3. 받아온 값을 변수에 저장
4. 광고 사진은 multipartformdata adPhotoInsert dao 에 들어가 저장
   광고 옵션은 opinsert dao 에 들어가 저장
   이외 나머지 값들은 adinsert dao 에 저장
5. 입력이 실패없이 완료될 시 트레이너 페이지로 화면전환

## ■ DAO & MAPPER

```
mDao.adPhotoInsert(map) - 광고 사진 등록
mDao.opinsert(opMap) - 광고 옵션 등록
mDao.adinsert(adadd) - 위 이외의 값들 등록
```

```
<insert id="adinsert" parameterType="question" useGeneratedKeys="true" keyProperty="xxx" >
<selectKey keyProperty="xxx" resultType="String" order="AFTER">
                select ad_seq.currval from dual
</selectKey>
INSERT INTO AD (AD_CODE,AD_NAME,AD_TITLE,AD_CATEGORY,AD_STATUS,AD_DATE,AD_CONTENT)
VALUES(AD_SEQ.NEXTVAL,#{ad_name},#{ad_title},#{ad_category},#{ad_status},SYSDATE,#{ad_content})
      </insert>
```

1. 현재 광고 번호의 값을 xxx 로 받아 둔다
2. 받아둔 xxx 번호의 다음번호로 각 값들을 입력한다

```
<insert id="adPhotoInsert" parameterType="HashMap">
        INSERT INTO AP VALUES(#{ap_adcode},#{ap_image})
      </insert>
```

1. 해당 광고 코드와 이미지를 관리하는 테이블에 값들을 저장한다

```
<insert id="opinsert">
INSERT INTO OP VALUES(OP_SEQ.NEXTVAL,#{op_adcode},#{op_trainer},#{op_price}
,#{op_period},#{op_times},#{op_day},#{op_content},#{op_clock},#{op_personnel},0)
        </insert>
```

1. 광고 옵션 번호를 기준으로 옵션 번호의 값을 순차적으로 입력한다.

## ■ FRONT-END

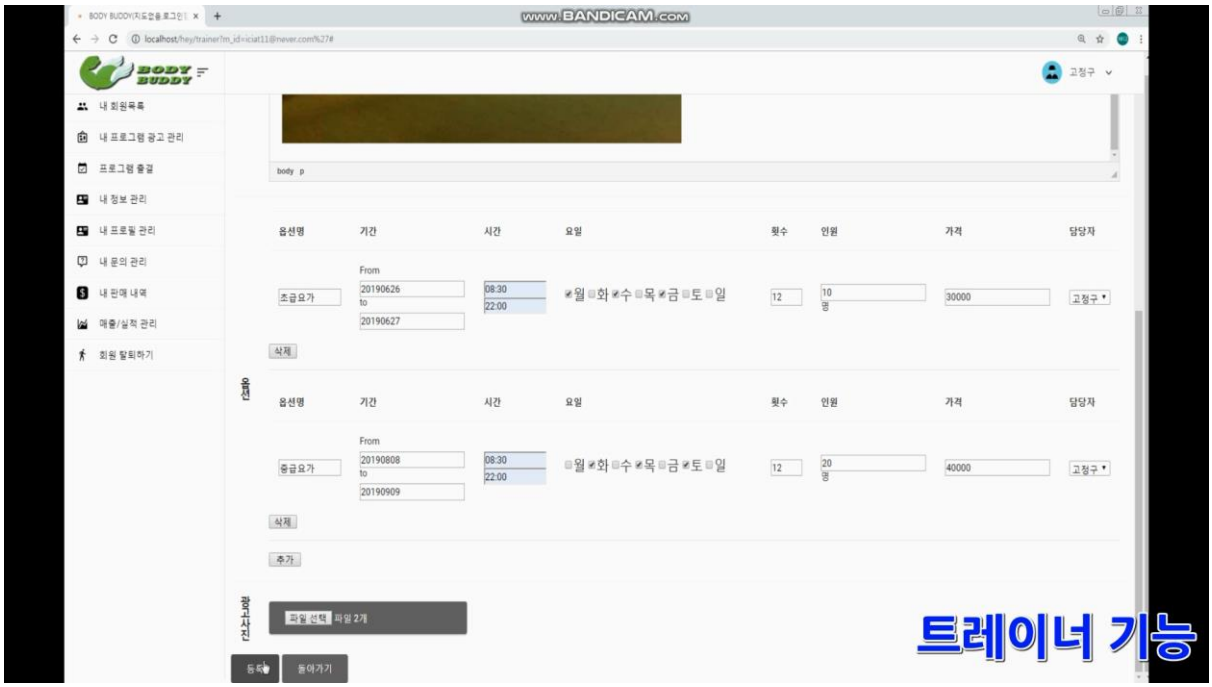
```
<form id="frm" action="adinsert" method="post" enctype="multipart/form-data">
    <div class="container">
        <table id="recent-purchases-listing" class="table">
            <tbody>
              <tr>
                <th style="width: 200px">분류</th>
                    <td style="vertical-align: middle">
                      <select id="ex-select" name="ad_category" onchange="selectExercise()"
                                class="form-control"    style="width: 150px;">
                        <option class="nothing" value="nothing" >선택해주세요</option>
                        <option class="fitness" value="normal">일반</option>
```

```
      <option class="pt" value="pt">피트니스-PT</option>
      <option class="homeTraining" value="homeTraining">홈트레이닝</option>
      <option class="pilates" value="pilates">필라테스</option>
      <option class="yoga" value="yoga">요가</option>
    </select>
  </td>
</tr>
<tr>
  <th>제목</th>
    <td style="vertical-align: middle"><input type="text"
            name="ad_title" style="height: 40px; width: 600px">
    </td>
</tr>
<tr>
  <th>내용</th>
    <td style="vertical-align: middle">
      <textarea id="ad_content" name="ad_content"></textarea>
    </td>
  </tr>
  <tr id="time" >
    <th>옵션</th>
      <td style="vertical-align: middle">
        <div class="radio">
          <div id="pre_set" class="pre_set"><!-- style="display: none" -->
            <table id="optionTable">
              <tr>
                <th>옵션명</th>
                <th>기간</th>
                <th>시간</th>
                <th class="programOption">요일</th>
                <th class="programOption">횟수</th>
                <th class="programOption">인원</th>
                <th>가격</th>
                <th class="programOption">담당자</th>
              </tr>
              <tr>
                <td><input type="text" name="op_content" style="width: 100px"
                           placeholder="옵션명" class="reset">
                </td>
                <td><label for="dates" class="normalOption">일 수(Dates)</label>
                    <input type="text"  id="dates" name="op_period" class="dates normalOption">
                    <label for="from" class="programOption">From</label>
                    <input type="text"  id="from" name="op_period1" class="from programOption">
                    <label for="to" class="programOption">to</label>
                    <input type="text" id="to" name="op_period2" class="to programOption">
                 </td>
                 <td><input type="text" name="op_clock1" style="width: 100px"
                            placeholder="시작시간" class="reset">
                      <input type="text" name="op_clock2" style="width: 100px"
                            placeholder="종료시간" class="reset">
                  </td>
                           <td class="programOption">
                      <div class="checkbox" style="width: 300px; font-size: 20px" class="reset">
                        <label for="foo1">
                            <input type="checkbox" id="foo1" name="day" value="월 ">월
                                 </label>
                        <label for="foo2">
                            <input type="checkbox" id="foo2" name="day" value="화 ">화
                        </label>
                        <label for="foo3">
                            <input type="checkbox" id="foo3" name="day" value="수 ">수
```

```
                                    </label>
                                    <label for="foo4">
                                        <input type="checkbox" id="foo4" name="day" value="목 ">목
                                    </label>
                                    <label for="foo5">
                                        <input type="checkbox" id="foo5" name="day" value="금 ">금
                                    </label> <label for="foo6">
                                        <input type="checkbox" id="foo6" name="day" value="토 ">토
                                    </label>
                                    <label for="foo7">
                                        <input type="checkbox" id="foo7" name="day" value="일 ">일
                                    </label>
                                    <label for="foo7">
                                        <input type="hidden" id="foo7" name="day" value="@ ">
                                    </label>
                                  </div>
                                </td>


                                <td class="programOption">
                                    <input type="text" name="op_times" style="width: 50px"
                                           placeholder="횟수" class="reset">
                                </td>
                                <td class="programOption">
                                    <input type="text" name="op_personnel"
                                           placeholder="모집인원" class="reset">명
                                </td>
                                <td>
                                    <input type="text" name="op_price" placeholder="가격(원)" class="reset">
                                </td>
                                <td class="programOption">${trainerlist}</td><!-- 담당자 -->
                              </tr>
                            </table>
                            <input type="button" value="삭제" onclick="remove_item(this)"><hr>
                          </div>
                          <div id="field"></div>
                          <input type="hidden" value="0" name="checkNum" id="checkNum">
                          <input type="button" value=" 추가 " onclick="add_item()"><br>
                                    </div>
                        </td>
                      </tr>
                      <tr>
                        <th>광고사진</th>
                        <td><input type="file" class="btn btn-secondary pull-left"
                                   name="ap_image" multiple="multiple"></td>
                      </tr>
                    </tbody>
                  </table>
                  <button id="ins" class="btn btn-secondary" type="submit">등록</button>
                  <button id="back" class="btn btn-secondary" type="button">돌아가기</button>
                </div>
              </form>
```

1. 담당자 란에 등록 가능한 트레이너의 목록을 불러온다. 개인트레이너는 자신의 이름이 선택된다
2. 조건에 따라 필요한 입력값을 입력
3. 입력이 완료되면 등록 완료

## ■ 결과


www.BANDICAM.com
BODY BUDDY
고정구
내 회원목록
내 프로그램 광고 관리
프로그램 홍보
내 정보 관리
내 프로필 관리
내 문의 관리
내 판매 내역
매출/실적 관리
회원 탈퇴하기
옵션
옵션명
기간
시간
요일
횟수
인원
가격
담당자
From
초급요가
20190626
to
20190627
08:30
22:00
월 화 수 목 금 토 일
12
10
명
30000
고정구
삭제
From
중급요가
20190808
to
20190909
08:30
22:00
월 화 수 목 금 토 일
12
20
명
40000
고정구
삭제
추가
광고사진
파일 선택 파일 2개
등록
돌아가기
트레이너 기능

## 15 JOB_NAME : 광고 수정 Form 가져오기 작성자 : 박한솔

### ■ 개요

1. 클라이언트가 등록한 광고중 수정할내용을 가져온다 .
2. 기존 광고 정보 가져와서 수정 폼에 출력

### ■ CONTROLLER

(코드)

```
@ResponseBody
@RequestMapping(value = "/advertisemodifyfrm", method = RequestMethod.POST)
public ModelAndView advertiseModifyFrm(String ad_code) {
        mav = mm.advertiseModifyFrm(ad_code);
        return mav;
}
```

### ■ SERVICE

(코드)

```
public ModelAndView advertiseModifyFrm(String ad_code) {
        // int i = mDao.opcount(ad_code);
        mav = new ModelAndView();
        Member mb = (Member) session.getAttribute("mb");
        String id = mb.getM_id();
        ArrayList<Question> opList = mDao.advertiseModifyFrm(ad_code);
        List<YesOrNo> yn = mDao.trinerlist(id);
        String makeopHtml = opHtml(opList, opList.size(), yn);
        Question list = new Question();
        String content = opList.get(0).getAd_content();
        mav.addObject("opHtml", makeopHtml);
        mav.addObject("opList", opList);
        mav.addObject("content", content);
        mav.setViewName("manage/advertisemodifyfrm");
        return mav;
}
```

(설명)

가져온 광고 코드를 이용하여 정보를 가져와 `makeopHtml, opList, content` 담아서 출력한다

### ■ DAO & MAPPER

(코드)

```
public ArrayList<Question> advertiseModifyFrm(String ad_code);
<select id="advertiseModifyFrm" parameterType="String"
        resultType="question">
        SELECT * FROM OP
        LEFT JOIN AD
        ON AD_CODE = OP_ADCODE
        WHERE OP_ADCODE = #{ad_code}
</select>
public ArrayList<YesOrNo> trinerlist(String id);
<select id="trinerlist" resultType="yesorno" parameterType="String">
        SELECT M_NAME,T_ID FROM M LEFT JOIN T ON M_ID = T_ID WHERE T_CID = #{id}
</select>
```

### ■ FRONT-END

(코드)

```
$(".eBtn").each(function() {
        $(this).click(function() {
                var num = $(this).parents().eq(1).children().eq(1).val();
                console.log(num);
                $.ajax({
                        type : "post",
                        url : "advertisemodifyfrm",
                        data : {
```

```
                                        "ad_code" : num
                                    },
                                    dataType : "html",
                                    error : function() {
                                        alert('통신실패!!');
                                    },
                                    success : function(data) {
                                        /* $('#main').hide(); */
                                        $('#main').html(data);
                                    }
                                });
                            })
                        })
```

(설명)

num 에 광고 번호를 담아서 ajax 통신을 한다

## ■ 결과

광고 관리

| 카테고리 | 광고명 | 등록일 | 상태 | 관리 |
|---|---|---|---|---|
| 요가 | 요가 | 2019-06-27 20:01:00.0 | 모집중 | 수정 삭제 |

신규 등록

### 수정할 광고를 선택한다

분류 요가

제목 요가

내용

요가1111

| 옵션명 | 기간 | 시간 | 요일 | 횟수 | 인원 | 가격 | 담당자 |
|---|---|---|---|---|---|---|---|
| 중급요가 | 일 수(Dates) From 20190808 to 20190909 | 08:30 22:00 | □월 □화 ☑수 □목 □금 □토 □일 | 12 | 20 명 | 40000 | 고정구 |
| 초급요가 | 일 수(Dates) From 20190808 to 20190909 | 08:30 22:00 | □월 □화 ☑수 □목 □금 □토 □일 | 12 | 10 명 | 30000 | 고정구 |

수정 돌아가기

## 16 JOB_NAME : 광고 수정하기 작성자 : 박한솔, 장기림

### ■ 개요

1. 변경 불가능한 정보(광고 제목, 카테고리, 작성자 등) 외 모든 정보 수정 가능
2. 광고 작성과 동일한 검증과정 진행

### ■ CONTROLLER

```
@RequestMapping(value = "/admodify", method = RequestMethod.POST)
        public ModelAndView adeModify(MultipartHttpServletRequest multi) {
                mav = mm.adModify(multi);
                return mav;
        }
```

사진 업로드가 포함되어 있기에 멀티파트를 사용한다

### ■ SERVICE

```
public ModelAndView adModify(MultipartHttpServletRequest multi) {
                mav = new ModelAndView();
                Question adadd = new Question();
                List<String> ad_nameTName = new ArrayList<>();
                String ad_code = null;
                String ad_category = null;
                String ad_title = null;
                String ad_content = null;
                String xxx = null;
                String[] op_contentValues = null;
                String[] day1 = null;
                Map<String, String> opMap = new HashMap<String, String>();
                List<String> ad_nameTId = null;
                String[] op_trainerValues = null;
                String[] dayValues = null;
                String[] op_periodValues = null;
                String[] op_period_1sValues = null;
                String[] op_period_2sValues = null;
                String[] op_timesValues = null;
                String[] op_clock1Values = null;
                String[] op_clock2Values = null;
                String[] op_priceValues = null;
                String[] op_personnelValues = null;
                String[] op_codeValues = null;
                Member mb = (Member) session.getAttribute("mb");
                adadd.setAd_name(mb.getM_id());/
                opMap.put("ad_name", mb.getM_id());
                ad_category = multi.getParameter("ad_category");
                if (ad_category != null && ad_category != "") {
                        if (ad_category.equals("normal")) {
                                adadd.setAd_category("일반");
                                opMap.put("ad_category", "일반");
                        }
                        if (ad_category.equals("pt")) {
                                adadd.setAd_category("피트니스");
                                opMap.put("ad_category", "피트니스");
                        }
                        if (ad_category.equals("homeTraining")) {
                                adadd.setAd_category("홈트레이닝");
                                opMap.put("ad_category", "홈트레이닝");
                        }
                        if (ad_category.equals("pilates")) {
                                adadd.setAd_category("필라테스");
                                opMap.put("ad_category", "필라테스");
                        }
                        if (ad_category.equals("yoga")) {
```

```
                    adadd.setAd_category("요가");
                    opMap.put("ad_category", "요가");
            }
}

ad_title = multi.getParameter("ad_title");
if (ad_title != null && ad_title != "") {
            adadd.setAd_title(ad_title);
            opMap.put("ad_title", ad_title);
}
ad_content = multi.getParameter("ad_content");
if (ad_content != null && ad_content != "") {
            adadd.setAd_content(ad_content);
            opMap.put("ad_content", ad_content);
}
op_codeValues = multi.getParameterValues("op_code");
op_contentValues = multi.getParameterValues("op_content");
op_timesValues = multi.getParameterValues("op_times");
op_clock1Values = multi.getParameterValues("op_clock1");
op_clock2Values = multi.getParameterValues("op_clock2");
op_priceValues = multi.getParameterValues("op_price");
op_personnelValues = multi.getParameterValues("op_personnel");
xxx = multi.getParameter("ad_code");
op_trainerValues = multi.getParameterValues("op_trainer");
if (mb.getM_kind().equals("t")) {
            ad_nameTId = new ArrayList<>();
            for (int i = 0; i < op_trainerValues.length; i++) {
                    if (op_trainerValues[i] != "") {
                            String[] ad_nameT = op_trainerValues[i].split(",");
                            ad_nameTId.add(ad_nameT[1]);
                            ad_nameTName.add(ad_nameT[0]);
                    }
            }
}
dayValues = multi.getParameterValues("day");
String b = "";
for (int i = 0; i < dayValues.length; i++) {
            if (dayValues[i] != "") {
                    b += dayValues[i];
            }
}
System.out.println("b=" + b);
day1 = b.split(" ");
op_periodValues = multi.getParameterValues("op_period");
if (op_periodValues.length != 0 && op_periodValues != null) {
            for (int i = 0; i < op_periodValues.length; i++) {
            }
}
op_period_1sValues = multi.getParameterValues("op_period1");
if (op_period_1sValues.length != 0 && op_period_1sValues != null) {
            for (int i = 0; i < op_period_1sValues.length; i++) {
            }
}
op_period_2sValues = multi.getParameterValues("op_period2");
if (op_period_2sValues.length != 0 && op_period_2sValues != null) {
            for (int i = 0; i < op_period_2sValues.length; i++) {
            }
}
adadd.setAd_status("모집중");
opMap.put("ad_status", "모집중");
adadd.setXxx(xxx);
adadd.setAd_code(xxx);
opMap.put("op_adcode", xxx);
opMap.put("ad_code", xxx);
if (mDao.admodify(adadd)) {
            String root = multi.getSession().getServletContext().getRealPath("/");
            String path = root + "resources/upload/";
            File dir = new File(path);
            if (!dir.isDirectory()) {
```

```
                        dir.mkdirs();
                }
                Map<String, String> map = new HashMap<>();
                List<MultipartFile> files = multi.getFiles("ap_image");
                String oriFileName = null;
                String sysFileName = null;
                try {
                        for (int i = 0; i < files.size(); i++) {
                                oriFileName = files.get(i).getOriginalFilename();
                                String index = String.valueOf(i);
                                sysFileName = index + System.currentTimeMillis() + "."
                                + oriFileName.substring(oriFileName.lastIndexOf(".") + 1);
                                map.put("ap_adcode", xxx);
                                map.put("ap_image", sysFileName);
                                if (sysFileName != null) {
                                files.get(i).transferTo(new File(path + sysFileName));
                                        mDao.deletepf(m.getM_id());
                                        if (mDao.adpfmodify(map)) {
                                        }
                                }
                        }
                } catch (Exception e) {
                }
                for (int i = 0; i < op_contentValues.length; i++) {
                        opMap.put("op_code", op_codeValues[i]);
                        opMap.put("op_adcode", xxx);
                        opMap.put("op_content", op_contentValues[i]);
                        opMap.put("op_price", op_priceValues[i]);
                        if (ad_category.equals("normal")) {
                                opMap.put("op_period", op_periodValues[i]);
                                opMap.put("op_personnel", "-100");
                                opMap.put("op_times", op_timesValues[i]);
                                opMap.put("op_day", day1[i]);
                        } else {
                        if (op_period_1sValues[i] != "" && op_period_2sValues[i] != "") {
                opMap.put("op_period", op_period_1sValues[i] + "~" + op_period_2sValues[i]);
                                }
                                if (op_timesValues[i] != "") {
                                        opMap.put("op_times", op_timesValues[i]);
                                }
                                if (day1 != null && day1.length != 0 && day1[i] != "") {
                                        opMap.put("op_day", day1[i]);
                                }
                                if (op_personnelValues[i] != "") {
                                opMap.put("op_personnel", op_personnelValues[i]);
                                }
                        }

                        if (op_clock1Values[i] != "" && op_clock2Values[i] != "") {
                opMap.put("op_clock", op_clock1Values[i] + "~" + op_clock2Values[i]);
                        }
                        adadd.setAd_code(xxx);
                        if (mb.getM_kind().equals("t")) {
                                if (ad_nameTId.size() != 0)
                                        opMap.put("op_trainer", ad_nameTId.get(i));
                        } else if (mb.getM_kind().equals("c")) {
                                for (int s = 0; s < op_trainerValues.length; s++) {
                                        if (op_trainerValues[s] != "") {
                                String[] ad_nameT = op_trainerValues[s].split(",");
                                        ad_nameTName.add(ad_nameT[0]);
                }
                                }
                        }
                        if (op_trainerValues[i] != "") {
                                opMap.put("op_trainer",ad_nameTId.get(i));
                        }
                        if (mDao.opmodify(opMap)) {
                        }
                }
        } else {
        }
```

```
            if (mb.getM_kind().equals("t")) {
                    mav.setViewName("forward:/");
            } else if (mb.getM_kind().equals("c")) {
                    mav.setViewName("forward:/");
            }
            return mav;
    }
```

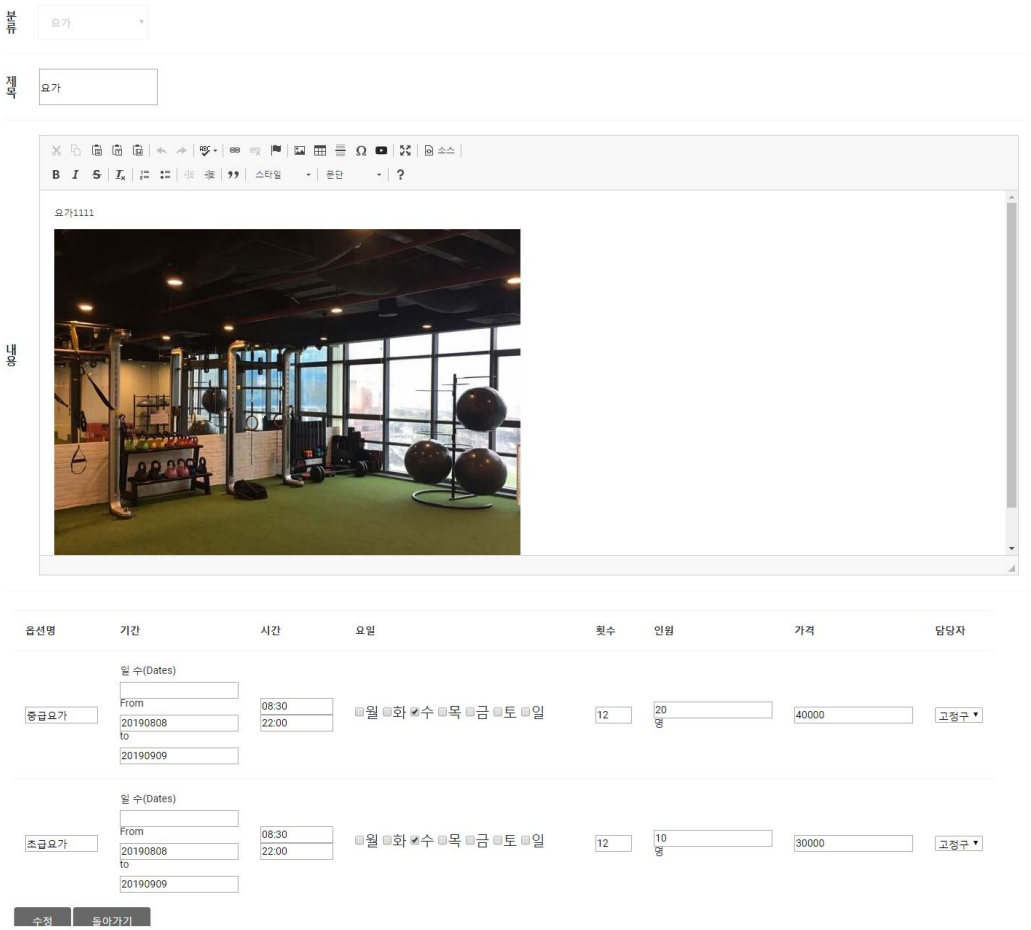
가져온 값을 비교하여 PUT 을 이용하여 일치하는 값을 opMap 에 넣어준다

## ■ DAO & MAPPER

```
public boolean admodify(Question adadd);

<update id="admodify" parameterType="question">
UPDATE AD SET AD_TITLE = #{ad_title} ,AD_CONTENT=#{ad_content}, AD_CATEGORY =#{ad_category},
AD_STATUS ='모집중',AD_DATE=sysdate
where ad_code = #{xxx}
</update>

public boolean opmodify(Map<String, String> opMap);
<update id="opmodify" parameterType="HashMap">
UPDATE OP SET OP_TRAINER=#{op_trainer} ,OP_PRICE=#{op_price},OP_PERIOD=#{op_period},OP_TIMES=#{op_times},
OP_DAY=#{op_day},OP_CONTENT=#{op_content},OP_CLOCK=#{op_clock},OP_PERSONNEL=#{op_personnel},OP_EXPIRE='0'
WHERE OP_CODE=#{op_code}
</update>


public boolean adpfmodify(Map<String, String> map);
<insert id="adpfmodify" parameterType="hashmap">
INSERT INTO ADPHOTO VALUES (#{ap_adcode},#{ap_image})
</insert>


public boolean deletepf(String ad_code);
<delete id="deletepf" parameterType="String">
delete from adphoto where ap_adcode=#{ad_code}
</delete>
```

수정된 광고 내용을 업데이트 한후, 광고 옵션을 업데이트, 사진갯수 변동이있을수도 있기 때문에 전부 삭제후 다시 INSERT 해준다.

## ■ 결과




**정보를 수정후 수정버튼을 누르면 변경된 내용이 저장된다**

## 17 JOB_NAME : 광고 삭제 작성자 : 박한솔, 장기림

### ■ 개요

1. 데이터 보존을 위해 완전삭제가 아닌 해당 광고를 만료시키게 된다.

### ■ CONTROLLER

(코드)

```
@ResponseBody
        @RequestMapping(value = "/deletead" ,method = RequestMethod.POST)
        public String deleteAd(String ad_code) {
                String str =mm.deleteAd(ad_code);
                return str;
        }
```

(설명)

삭제할 광고번호를 가져와서 deletrAd 에 넘긴다

### ■ SERVICE

(코드)

```
public String deleteAd(String ad_code) {
                Integer success1 = mDao.deleteAdphoto(ad_code);
                Integer success = mDao.deleteAd(ad_code);
                Gson gson = new Gson();
                String str = gson.toString();
                return str;
        }
```

(설명)

ad_code 를 넘긴다 ajax 통신이기에 반환값을 string 을 넘겨준다

### ■ DAO & MAPPER

(코드)

```
public Integer deleteAdphoto(String ad_code);
public Integer deleteAd(String ad_code);

        <update id="deleteAd" parameterType="String">
                UPDATE AD SET AD_STATUS ='만료됨' WHERE AD_CODE=#{ad_code}
        </update>
        <delete id="deleteAdphoto" parameterType="String">
        DELETE FROM ADPHOTO WHERE AP_ADCODE = #{ad_code}
        </delete>
```

(설명)

가져온 ad_code 값을 넘겨 해당하는 광고

### ■ FRONT-END

(코드)

```
$(".dBtn").each(function() {
        $(this).click(function() {
                var num = $(this).parents().eq(1).children().eq(1).val();
                console.log(num);
                var id = "#" + num;
                console.log(id);
                $.ajax({
                        url : "deletead",
                        type : "post",
```

```
                    data : {
                        "ad_code" : num
                    },
                    dataType : "html",
                    success : function(data) {
                        console.log(data)
                        $(id).prop("style", "display: none");
                    },
                    error : function(error) {
                        console.log(error);
                    }
                })
            });//end ajax
```

(설명)
변수 num 에 선택한 광고에 광고번호를 담아 ajax 통신후 성공시 해당 광고를 디스플레이로 임시로 가려준다

## ■ 결과

광고 관리

| 카테고리 | 광고명 | 등록일 | 상태 | 관리 |
|---|---|---|---|---|
| 필라테스 | 필라테스 | 2019-06-27 19:56:07.0 | 모집중 | 수정 삭제 |
| 요가 | 요가 | 2019-06-27 20:01:00.0 | 모집중 | 수정 삭제 |

신규 등록

광고 삭제 버튼을 누르면 광고가 바로 만료가되어 출력되지않는다

광고 관리

| 카테고리 | 광고명 | 등록일 | 상태 | 관리 |
|---|---|---|---|---|
| 요가 | 요가 | 2019-06-27 20:01:00.0 | 모집중 | 수정 삭제 |

신규 등록

## 18 JOB_NAME : 관리자 페이지 메인 대시보드 작성자 : 윤상기, 권순철

■ 개요

1. 업체 회원으로 로그인을 하게되면 관리자 페이지로 이동이 가능하고 첫 기본 페이지에 최근 신규가입, 판매내역, 등록중인 광고, 매출과 실적을 간단하게 보여주게 됩니다.

■ CONTROLLER

```
@RequestMapping(value = "/company")
public ModelAndView company(HttpServletRequest request) {
        mav = new ModelAndView();
        Member mb = (Member) session.getAttribute("mb");
        session.setAttribute("mb", mb);
        mav.addObject("session_id", mb.getM_id());

        System.out.println("mb.getM_id() = "+mb.getM_id());
        System.out.println("mb.getM_kind() = "+mb.getM_kind());
        System.out.println("mb.getM_name() = "+mb.getM_name());

        mav = ks.mainList(mb);

        return mav;
}
```

1. Session 으로 받아온 mb 를 Member Bean 에 담아 mainList(mb)의 파라미터로 넘겨 준다.

■ SERVICE

```
public ModelAndView mainList(Member mb) {
        mav = new ModelAndView();
        String id = mb.getM_id();
        int j = 5;

        ArrayList<Member> mList = new ArrayList<Member>();
        ArrayList<Member> mList1 = new ArrayList<Member>();
        try {
                System.out.println("getMainMemberList mDao in");
                mList1 = ksDao.getMainMemberList(id);
                System.out.println("getMainMemberList mDao out" + mList1.size());
                if (mList1.size() < 5) {
                        j = mList1.size();
                }
                if (0 != mList1.size()) {
                        System.out.println("getMainMemberList if in");
                        for (int i = 0; i < j; i++) {
                                System.out.println("getMainMemberList for in");
                                mList.add(i, mList1.get(i));
                                System.out.println(mList.get(i).getM_id());
                        }
                        System.out.println("getMainMemberList for out");
                        j = 5;
                }
                System.out.println("member list select success");
                mav.addObject("mList", mList);
        } catch (Exception e) {
                System.out.println("member list select error");
        }


        ArrayList<Sales> sList = new ArrayList<Sales>();
        ArrayList<Sales> sList1 = new ArrayList<Sales>();
        try {
                System.out.println("getSalesHistory mDao in");
                sList1 = sDao.getMainSalesHistory(id);
                if (sList1.size() < 5) {
                        j = sList1.size();
                }
                if (0 != sList1.size()) {
                        for (int i = 0; i < j; i++) {
```

```
                                sList.add(i, sList1.get(i));
                                // sList.set(i, sList1.get(i));
                                System.out.println(sList.get(i).getPs_code());
                        }
                        j = 5;
                }
                System.out.println("getSalesHistory select success");
                mav.addObject("sList", sList);
        } catch (Exception e) {
                System.out.println("getSalesHistory list select error");
        }


        ArrayList<Question> aList = new ArrayList<Question>();
        ArrayList<Question> aList1 = new ArrayList<Question>();
        try {
                System.out.println("getMainAdvertise mDao in");
                aList1 = ksDao.getMainAdvertise(id);
                if (aList1.size() < 5) {
                        j = aList1.size();
                }
                if (0 != aList1.size()) {
                        for (int i = 0; i < j; i++) {
                                aList.add(i, aList1.get(i));
                                // sList.set(i, sList1.get(i));
                                System.out.println(aList.get(i).getAd_code());
                        }

                }
                ////////////////////////////////
                        List<Map<String, String>> getSalesList = null;
                        List<Map<String, String>> getSalesAllList = null;
                        List<Map<String, String>> getSalescList = null;
                        List<Map<String, String>> getSalesAllc = null;
                        Member sessionMb = (Member) session.getAttribute("mb");
                        String m_id=sessionMb.getM_id();
                        String ad_code=yDao.getAd_code(m_id);
                        Map<String,String> cl = new HashMap<>();
                        cl.put("ad_code", ad_code);
                        cl.put("m_id", m_id);
                        getSalesList = yDao.getsales(cl);
                        getSalesAllList = yDao.getsalesAll(cl);
                        getSalescList = yDao.getSalescList(m_id);
                        getSalesAllc = yDao.getSalesAllcList(m_id);
                        //getSalescList = yDao.getsalesAlla(m_id);
                        String html = ys.makeHTMLsalesList(getSalesList);
                        String html2 = ys.makeHTMLsalesAllList(getSalesAllList);
                        String html3 = ys.makeHTMLsalescList(getSalescList);
                        String html4 = ys.makeHTMLsalesAllcList(getSalesAllc);
                        mav.addObject("getSalesList", html);
                        mav.addObject("getSalesAllList", html2);
                        mav.addObject("getSalescList", html3);
                        mav.addObject("getSalesAllcList", html4);
                ////////////////////////////////////////////////////////

                System.out.println("getMainAdvertise select success");
                mav.addObject("aList", aList);
        } catch (Exception e) {
                System.out.println("getMainAdvertise list select error");
        }

        mav.setViewName("manage/company/companyMain");
        return mav;
}
```

1. **getMainMeberList(id)로 id 파라미터를 넘겨 가입된 회원 리스트를 Member ArrayList 타입에 mList 에 담는다.**
2. **getMainSalesHistory(id)로 id 를 파라미터로 넘겨 판매된 내역을 Salse ArrayList 타입인 sList 에 담는다.**
3. **getMainAdvertise(id)로 id 를 파라미터로 넘겨 현재 광고중인 광고를 Question ArrayList 타입인 aList 에 담는다.**
4. **getAd_code(m_id)로 m_id 를 파라미터로 넘겨 광고 코트를 받아온다.**
5. **getSales(cl)로 HashMap cl 에 ad_code 와 m_id 를 담아 파라미터로 cl 을 넘겨 트레이너의 실적을 받아온다.**

6. getSalesAll(cl)로 cl 파라미터를 넘겨 트레이너의 총 실적을 받아온다.
7. getSalescList(m_id)로 m_id 를 파라미터로 넘겨 카테고리별 광고의 판매 실적을 받아온다.
8. getSalesAllcList(m_id)로 m_id 를 파라미터로 넘겨 총 카테리고별 판매 실적을 담는다.
9. makeHTMLsalesList(getSalesList),makeHTMLsalesAllList(getSalesAllList),makeHTMLsalescList(getSalescList),
10. makeHTMLsalesAllcList(getSalesAllc)를 사용하여 HTML 로 변환한다.
11. companyMain.jsp 페이지로 이동한다.

## ■ DAO & MAPPER

```
ArrayList<Member> getMainMemberList(String id);
ArrayList<Sales> getMainSalesHistory(String id);
ArrayList<Question> getMainAdvertise(String id);
String getAd_code(String m_id);
List<Map<String, String>> getsales(Map<String, String> cl);
List<Map<String, String>> getsalesAll(Map<String, String> cl);
List<Map<String, String>> getSalescList(String m_id);
List<Map<String, String>> getSalesAllcList(String m_id);
```

**(1)DAO**

```
---------------------------------------------------------------------------------------
<select id="getMainMemberList" resultType="member">
        select distinct rownum,m.m_id,m.m_name,m.m_phone,to_char(m_birth,'YYYY.MM.DD') as
        m_birth,ps_date from m left join ps on m.m_id=ps.ps_mid left join op on
        ps.ps_adcode=op.op_adcode left join ad on ad.ad_code=op.op_adcode where
        ad.ad_name = #{id}
 </select>
```

**넘겨받은 id 를 사용하여 가입된 회원 전체를 SELECT 한다.**

```
<select id="getMainSalesHistory" resultType="sales">
        Select ps.ps_code,ad.ad_title,m.m_id,m.m_name,ps.ps_price,ps.ps_date,ps.ps_adcode
        from m left join ps on ps.ps_mid= m.m_id left join ad on ps.ps_adcode
        = ad.ad_code where ad.ad_name = #{id} order by ps_code desc
 </select>
```

**넘겨받은 id 를 사용하여 판매 내역 전체를 SELECT 한다.**

```
<select id="getMainAdvertise" resultType="question">
        select * from ad where ad_name = #{id} and ad_status != '만료됨'
 </select>
```

**넘겨받은 id 를 이용하여 만료되지 않은 광고 전체를 SELECT 한다.**

```
<select id="getAd_code" parameterType="String" resultType="String">
        <![CDATA[ SELECT AD_CODE FROM AD WHERE AD_NAME=#{m_id} AND ROWNUM=1 ]]>
 </select>
```

**넘겨받은 m_id 를 사용하여 광고코드를 담는다.**

```
<select id="getsales" parameterType="String" resultType="HashMap">
  <![CDATA[SELECT T_CID,T_ID,M_NAME,OP_CONTENT,COUNT(PS_OPCODE) as PS_OPCODE,OP_PRICE,SUM(PS_PRICE) as PS_PRICE
  FROM M JOIN T ON M_ID=T_ID
  JOIN C ON C_ID=T_CID JOIN AD ON AD_NAME=C_ID JOIN OP ON OP_TRAINER= T_ID
  JOIN PS ON OP_CODE = PS_OPCODE WHERE T_CID=#{m_id} AND AD_CODE=#{ad_code}
  GROUP BY T_CID,M_NAME,T_ID,OP_CONTENT,OP_PRICE ]]>
 </select>
```

**트레이너별로 실적을 받아온다.**

```
<select id="getSalesAllcList" parameterType="String" resultType="HashMap">
   <![CDATA[ SELECT SUM(PS_PRICE) as PS_PRICE FROM M
    JOIN AD ON M_ID=AD_NAME JOIN OP ON AD_CODE=OP_ADCODE JOIN PS ON PS_OPCODE=OP_CODE
    WHERE M_ID=#{m_id} ORDER BY PS_PRICE DESC ]]>
</select>
```

**트레이너의 전체 실적을 받아온다.**

```
<select id="getSalescList" parameterType="String" resultType="HashMap">
<![CDATA[ SELECT AD_CATEGORY,OP_CONTENT,COUNT(PS_OPCODE) as PS_OPCODE,OP_PRICE,SUM(PS_PRICE) as PS_PRICE FROM M
JOIN AD ON M_ID=AD_NAME
JOIN OP ON AD_CODE=OP_ADCODE JOIN PS ON PS_OPCODE=OP_CODE WHERE M_ID=#{m_id}
GROUP BY AD_CATEGORY,OP_CONTENT,OP_PRICE ORDER BY PS_PRICE DESC ]]>
</select>
```

**광고 카테고리별로 판매 실적을 받아온다.**

```
<select id="getSalesAllcList" parameterType="String" resultType="HashMap">
<![CDATA[SELECT SUM(PS_PRICE) as PS_PRICE FROM M JOIN AD ON M_ID=AD_NAME
   JOIN OP ON AD_CODE=OP_ADCODE JOIN PS ON PS_OPCODE=OP_CODE WHERE M_ID=#{m_id}
   ORDER BY PS_PRICE DESC ]]>
</select>
```

광고 카테고리별 총 판매 실적을 받아온다.

(2)MAPPER

■ FRONT-END

```
$(document).ready(function() {var t_cid = '${mb.m_id}'; /* $("#testInput").val(); */
          console.log(t_cid)
        $.ajax({
                type : "post",
                  url : "chart",
                  data : {
                            t_cid : t_cid
                          },
                  dataType : "json",
                    success : function(result) {
                              console.log("result:", result);
                              var str = "";
                              var name = "";
                              var ctx = document.getElementById("myChart").getContext('2d');
                              var arr = {};
                              for (var i = 0; i < result.length; i++) {
                            str += String(result[i].PS_PRICE).split(9) + ",";
                                  name += String(result[i].M_NAME).split(7) + ",";
                              }
                              var arr2 = (name.substring(0, name.length - 1)).split(",");
                              arr = (str.substring(0,str.length - 1)).split(",");
                              var myChart = new Chart(ctx, {
                                 type : 'bar',
                                 data : {
                                        labels : arr2,
                                        datasets : [ {
                                        label : '# of Votes',
                                        data : arr,
                                        backgroundColor : [
                                        'rgba(255, 99, 132, 0.2)',
                                        'rgba(54, 162, 235, 0.2)',
                                        'rgba(255, 206, 86, 0.2)',
                                        'rgba(75, 192, 192, 0.2)',
                                        'rgba(153, 102, 255, 0.2)',
                                        'rgba(255, 159, 64, 0.2)' ],

                                    borderColor : [
                                      'rgba(255,99,132,1)',
                                      'rgba(54, 162, 235, 1)',
                                       'rgba(255, 206, 86, 1)',
                                       'rgba(75, 192, 192, 1)',
                                       'rgba(153, 102, 255, 1)',
                                       'rgba(255, 159, 64, 1)' ],

                                    borderWidth : 1

                          } ]
          },

          options : {
                    maintainAspectRatio : true,
                    scales : {
                    yAxes : [ {
                  ticks : {
                beginAtZero : true
                    }
                } ]
                    }
```

```
                    }
                    });
                    },
                    error : function(er) {
                        console.log(er);
                }
            });//실적 end
    //매출 start
    var t_cid2 = '${mb.m_id}';
    console.log("DDDDDDDDDDDD" + t_cid2)

        $.ajax({
            type : "post",
            url : "charttwo",
            data : {
                    t_cid : t_cid2
            },
            dataType : "json",
            success : function(result) {
            console.log("result:", result);

            var str = "";
            var name = "";
            var ctx = document.getElementById("myChart2").getContext('2d');
            var arr = {};
            for (var i = 0; i < result.length; i++) {
                   str += String(result[i].PS_PRICE).split(9) + ",";
                     += String(result[i].AD_CATEGORY).split(7) + ",";
                     }
                     console.log(name)
                     var arr2 = (name.substring(0,name.length - 1)).split(",");
                     var arr = (str.substring(0,str.length - 1)).split(",");
                 var myChart2 = new Chart(ctx,{ type : 'bar', data : {

                   labels : arr2,

                   datasets : [ {

                     label : '# of Votes',

                     data : arr,

                     backgroundColor : [

                                        'rgba(255, 99, 132, 0.2)',

                                        'rgba(54, 162, 235, 0.2)',

                                        'rgba(255, 206, 86, 0.2)',

                                        'rgba(75, 192, 192, 0.2)',

                                        'rgba(153, 102, 255, 0.2)',

                                        'rgba(255, 159, 64, 0.2)' ],

                     borderColor : [

                                        'rgba(255,99,132,1)',

                                        'rgba(54, 162, 235, 1)',

                                        'rgba(255, 206, 86, 1)',

                                        'rgba(75, 192, 192, 1)',

                                        'rgba(153, 102, 255, 1)',

                                        'rgba(255, 159, 64, 1)' ],
```

```
                borderWidth : 1

        } ]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
        options : {

        maintainAspectRatio : true,scales : {
        yAxes : [ {

                ticks : {

                beginAtZero : true

                        }
                } ]
        }
      }
        });

        },
        error : function(er) {
        console.log(er);

        }
        });
});// redey End

<div id="bodybody" style="left: 0px; top: 15px; position: relative; display: block;">
        <div style="top: 0px; left: 0px; position: absolute; width: 550px; height: 300px;">
          <div><h3 style="text-align: center;margin-top:20px;">최근 신규 가입</h3></div>
            <table class="table table-hover" style="table-layout: fixed; border-right: 1px solid #EAEAEA;
border-top:1px solid #EAEAEA; height: 335px;">
                <thead>
                        <tr>
                                <th>이름</th>
                                <th>아이디</th>
                                <th>가입날짜</th>
                        </tr>
                </thead>
                <tbody>
                        <c:forEach var="member" items="${mList }">
                            <tr><td><a href="#"
                    onclick="profileN('${member.m_id }')"><nobr>${member.m_name }</nobr></a></td>
                                <td><a href="#"
                     onclick="profileN('${member.m_id }')"><nobr>${member.m_id }</nobr></a></td>
                                                <td>${member.ps_date }</td>
                                        </tr>
                                </c:forEach>
                                <td>${member.ps_date }</td>
                        </tbody>
                        </table>

                </div>
                <div style="left: 550px; position: absolute; width: 550px; height: 300px; display: block;">
                <h3 style="text-align: center;margin-top:20px;">최근 판매 내역</h3>
                <table class="table table-hover" style="table-layout: fixed; height: 335px; border-right: 1px
solid #EAEAEA; border-top:1px solid #EAEAEA;">
                <thead>
                        <tr>
                                <!-- <th>판매번호</th> -->
                                <th>광고명</th>
                                <th>결제자</th>
                                <th style="width: 20%;">금액</th>
                                <th>판매일자</th>
                        </tr>
                </thead>
                <tbody>
```

```
                        <c:forEach var="sales" items="${sList }">
                                <tr>
<td style="text-align: center; text-overflow:ellipsis; overflow:hidden;"><nobr><a href="#"
onclick="advertisedetail(${sales.ps_adcode})">${sales.ad_title}</a></nobr></td>
        <td style=" text-overflow:ellipsis; overflow:hidden;"><nobr><a href="#"
        onclick="advertisedetail(${sales.m_id})">${sales.m_name }(${sales.m_id})</a></nobr></td>
        <td style="text-align: right; text-overflow:ellipsis;
overflow:hidden;"><nobr>${sales.ps_price }</nobr></td>
        <td style=" text-overflow:ellipsis; overflow:hidden;"><nobr>${sales.ps_date }</nobr></td>
                                                </tr>
                                        </c:forEach>
                                </tbody>
                        </table>
                </div>
                <div style="left: 1100px; position: absolute; width: 550px; height: 300px; display: block;">
        <h3 style="text-align: center;margin-top:20px;">등록중인 광고</h3>
        <table class="table table-hover"  style="table-layout: fixed; border-top:1px solid #EAEAEA;">
                <thead>
                        <tr>
                                <th>광고번호</th>
                                <th>광고명</th>
                                <th>등록일자</th>
                                <th>상태</th>
                        </tr>
                </thead>
                <tbody>
                        <tr>
                                <c:forEach var="ad" items="${aList }">
                                        <tr>
<td style="text-overflow:ellipsis; overflow:hidden;"><nobr>${ad.ad_code }</nobr></td>
<td style="text-overflow:ellipsis; overflow:hidden;"><nobr><a href="#"
onclick="profileN('${ad.ad_title }')">${ad.ad_title }</a></nobr></td>
<td style="text-overflow:ellipsis; overflow:hidden;"><nobr>${ad.ad_date}</nobr></td>
<td style="text-overflow:ellipsis; overflow:hidden;"><nobr>${ad.ad_status}</nobr></td>
                </tr>
                </c:forEach>
        </tr>
</tbody>
        </table></div>
</div>
```

makeHTML 로 받아온 데이터를 가지고 매출과 실적 차트를 사용하여 화면에 보여줍니다.
신규가입,판매내역,등록중인 광고를 받아 화면에 보여줍니다.

## ■ 결과

### 최근 신규 가입

| 이름 | 아이디 | 가입날짜 |
|---|---|---|
| 김수남 | yoontest@naver.com | 2019-06-27 15:47:06.0 |
| 윤상구 | yoontest4@never.com | 2019-06-28 09:31:43.0 |
| 정재형 | icia2@never.com | 2019-06-27 15:42:16.0 |
| 정민수 | icia3@never.com | 2019-06-27 15:42:33.0 |
| 김사랑 | icia4@never.com | 2019-06-27 15:42:44.0 |

### 최근 판매 내역

| 광고명 | 결제자 | 금액 | 판매일자 |
|---|---|---|---|
| 요가 입니다. | 권상우(icia1@ne... | 60000 | 2019-07-09 16:2... |
| 동네 최대 휘트니... | 윤상구(yoontest... | 300000 | 2019-06-28 09:3... |
| 동네 최대 휘트니... | 윤상기(18sanggi... | 500000 | 2019-06-27 20:5... |
| 동네 최대 휘트니... | 강동민(icia10@n... | 300000 | 2019-06-27 17:5... |
| 부산최대 휘트니... | 권상우(icia1@ne... | 70000 | 2019-06-27 17:4... |

### 등록중인 광고

| 광고번호 | 광고명 | 등록일자 | 상태 |
|---|---|---|---|
| 441 | 요가 입니다. | 2019-07-09 16:... | 모집중 |
| 395 | 부산최대 휘트... | 2019-06-27 15:... | 모집중 |
| 392 | 동네 최대 휘트... | 2019-06-27 15:... | 모집중 |

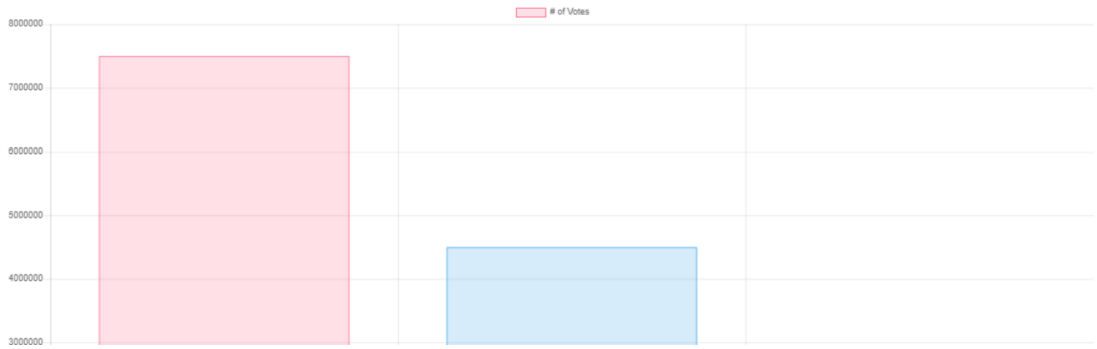
### 매출

| 분류 | 옵션명 | 총실적수 | 단가 | 총 금액 |
|---|---|---|---|---|
| PT | 노정근트레이너 | 5 | 500000 | 2500000 |
| PT | 제이팍트레이너 | 5 | 300000 | 1500000 |
| 일반 | 일반+케비넷 | 3 | 70000 | 210000 |
| 일반 | 일반옵션 | 3 | 50000 | 150000 |
| 요가 | 요가옵션 | 1 | 60000 | 60000 |
| 총 합계 | 4420000 | | | |

### 실적

| | | | | |
|---|---|---|---|---|
| 강정훈 | 노정근트레이너 | 5 | 500000 | 2500000 |
| 강성민 | 제이팍트레이너 | 5 | 300000 | 1500000 |
| 박한별 | 요가옵션 | 1 | 60000 | 60000 |

| 트레이너명 | 총실적수 | 총금액 |
|---|---|---|
| 강정훈 | 5 | 2500000 |
| 강성민 | 5 | 1500000 |
| 박한별 | 1 | 60000 |



## 19 JOB_NAME : 프로그램 별 회원 출결 관리 작성자 : 권순철

■ 개요

1. 일반 회원과 프로그램 회원이 출석을 하였을 때 출석 버튼을 눌러 회원의 출석을 입력 합니다.

■ CONTROLLER

```
@RequestMapping(value = "/normalcheck")
public ModelAndView normalCheckInsert(HttpServletRequest request) {
        mav = ks.normalCheckInsert(request);
        return mav;
 }
```

request 안에는 일반 회원의 결제 코드가 담겨 있습니다. normalCheckInsert()에 파라미터로 넘겨줍니다.

■ SERVICE

```
public ModelAndView normalCheckInsert(HttpServletRequest request) {
        String code = request.getParameter("code");

        mav = new ModelAndView();

        dc = new DailyCheck();

        boolean dci = false;
        int ps_date = 0;
        int ps_date1 = 0;
        int sd = 0;
        int check = 0;
        String da_code = null;
        String result = null;

        try {
                System.out.println("daily에 da_code 있는지 확인");
                da_code = ksDao.normalDailyCheckSelect(code);
                System.out.println("da_code1 = " + da_code);
                if (da_code != null) {
                        System.out.println("da_code가 있으면 실행");
                        dc = ksDao.normalDailyCheckSelect2(code);

                        check = ksDao.normalDailyCheckSelect3(dc);
                        System.out.println("check num = " + check);
                        if (check >= 1) {
                                result = "이미 출석 되었습니다.";
                                mav.setViewName("manage/company/programDailyCheck");
                                mav.addObject("result", result);
                                return mav;
                        }

                        sd = Integer.parseInt(dc.getSd()); // 현재 날짜 예)
                        String str = dc.getDa_opperiod(); // 출결조건 테이블에 실제 이용기간 예)
20190601~sysdate+13/20190618
                        ps_date = Integer.parseInt(str.substring(0, 8)); // ~기준으로 앞뒤로 자름
                        ps_date1 = Integer.parseInt(str.substring(9, 17));

                        if (ps_date <= sd || ps_date1 >= sd) {
                                dc.setStatus("이용중");
                                ksDao.normalDailyCheckUpdate(dc);
                                ksDao.normalDailyCheckInsert(dc);
                        } else if (ps_date > sd) {
                                dc.setStatus("대기중");
                                ksDao.normalDailyCheckUpdate(dc);
                        } else if (ps_date1 < sd) {
                                dc.setStatus("만료됨");
```

```
                        ksDao.normalDailyCheckUpdate(dc);
                }

        } else if (da_code == null) {
                System.out.println("da_code null");
                dc = ksDao.categoryCheck(code);
                dc.setPs_code(code);

                String category = dc.getAd_category();
                System.out.println("da_code null_category =" + category);

                if (category.equals("일반")) {
                        dc = ksDao.normalDailyCheckSelect1(code);
                        sd = Integer.parseInt(dc.getSd());
                        ps_date = Integer.parseInt(dc.getPs_date()); // 시작날짜
                        ps_date1 = Integer.parseInt(dc.getPs_date1()); // 종료날짜
                        String str1 = Integer.toString(ps_date) + "~" + Integer.toString(ps_date1);
                        System.out.println(str1);
                        if (ps_date <= sd && ps_date1 >= sd) {
                                System.out.println("이용중");
                                dc.setDate(str1);
                                dc.setStatus("이용중");
                                ksDao.normalDailyCheckInsert1(dc);
                                da_code = ksDao.normalDailyCheckSelect(code);
                                dc.setDa_code(da_code);
                                ksDao.normalDailyCheckInsert(dc);
                        } else if (ps_date > sd) {
                                System.out.println("대기중");
                                dc.setDate(str1);
                                dc.setStatus("대기중");
                                ksDao.normalDailyCheckInsert1(dc);
                        } else if (ps_date1 < sd) {
                                dc.setDate(str1);
                                dc.setStatus("만료됨");
                                ksDao.normalDailyCheckInsert1(dc);
                        }

                } else {
                        dc = ksDao.programDailyCheckSelect(code);
                        dc.setPs_code(code);
                        String str1 = dc.getOp_period();
                        sd = Integer.parseInt(dc.getSd());
                        ps_date = Integer.parseInt(str1.substring(0, 8));
                        ps_date1 = Integer.parseInt(str1.substring(9, 17));

                        if (ps_date <= sd && ps_date1 >= sd) {
                                System.out.println("이용중");
                                dc.setStatus("이용중");
                        System.out.println(dc.getPs_code() + dc.getStatus() + dc.getOp_period());
                                ksDao.programDailyCheckInsert(dc);
                                da_code = ksDao.normalDailyCheckSelect(code);
                                dc.setDa_code(da_code);
                                ksDao.normalDailyCheckInsert(dc);
                                } else if (ps_date > sd) {
                                System.out.println("대기중");
                                dc.setStatus("대기중");
                                ksDao.normalDailyCheckInsert1(dc);
                        } else if (ps_date1 < sd) {
                                dc.setStatus("만료됨");
                                ksDao.normalDailyCheckInsert1(dc);
                        }
                }

        }
```

```
                dci = true;
                System.out.println("normalDailyCheckInsert success");
        } catch (Exception e) {
                System.out.println("normalDailyCheckInsert fail");
        }

        if (dci) {
                System.out.println("normalCheckInsert success");
                result = "출석 입력 성공";

        } else {
                System.out.println("normalCheckInsert fail");
                result = "출석 입력 실패";
        }
        mav.setViewName("manage/company/programDailyCheck");
        mav.addObject("result", result);

        return mav;
}
```

normalDailyCheckSelect(code)를 이용하여 한번이라도 출석 체크를 했는지 확인합니다.
출석을 한번이라도 했다면 normalDailyCheckSelect2(code)사용하여 da_code 를 받아오고 normalDailyCheckSelect3(dc) 받아온 da_code 를 dc 파라미터로 넘겨 출석을 당일에 한적이 있는지 확인하여 check 가 1 이거나 이상이면 이미 출석했다고 보여주며 돌아갑니다.

Ps.date를 확인하여 현재날짜를 기준으로 이용중,대기중,만료됨을 확인하여 처리합니다. 이용중이라면 normalDailyCheckUpdate(dc)을 실행하여 이용중으로 상태를 변경합니다.

normalDailyCheckInsert(dc)을 실행하여 출석을 입력합니다.

출석을 한번도 한적이 없다면 일반인지 프로그램 회원인지 확인을 하여 처리 합니다.

일반 회원이라면 위와 같은 방법으로 ps_date를 이용하여 이용중,대기중,만료됨으로 상태 변경을 하여 출석을 입력합니다.

프로그램 회원도 마찬가지로 같은 방법을 사용 합니다.

## ■ DAO & MAPPER

```
String normalDailyCheckSelect(String code)
DailyCheck normalDailyCheckSelect2(String code)
int normalDailyCheckSelect3(DailyCheck dc)
void normalDailyCheckUpdate(DailyCheck dc)
void normalDailyCheckInsert(DailyCheck dc);
DailyCheck categoryCheck(String code);
DailyCheck normalDailyCheckSelect1(String code)
void normalDailyCheckInsert(DailyCheck dc)
void normalDailyCheckInsert1(DailyCheck dc)
```

(1)DAO

```
<select id="normalDailyCheckSelect" resultType="String" parameterType="String">
        select da_code from daily where da_pscode = #{code}
</select>
```

결제 코드를 사용하여 da_code 를 받아와서 da_code 가 있는지 확인을 합니다.

```
<select id="normalDailyCheckSelect2" resultType="dc" parameterType="String">
  Select da_code,da_pscode,da_status,da_opperiod,to_char(sysdate,'YYYYMMDD') as sd from daily where da_pscode
  = #{code}
</select>
```

결제 코드를 사용하여 테이블안에 정보를 받아 옵니다.

```
<select id="normalDailyCheckSelect3" parameterType="dc" resultType="int">
      <![CDATA[select count(*) from dailycheckn where dn_dacode = #{da_code} and dn_date= #{sd}]]>
</select>
```

오늘 날짜에 입력된 데이터가 있는지 확인합니다.

```
<update id="normalDailyCheckUpdate" parameterType="dc">
       update daily set da_status = #{status} where da_code = #{da_code}
</update>
```

Da_code 를 사용하여 현재 상태를 이용중,대기중,만료됨 으로 변경합니다.

```
<insert id="normalDailyCheckInsert" parameterType="dc">
        insert into dailycheckn values(#{da_code},to_char(sysdate,'YYYYMMDD'))
```

```
 </insert>
```

해당 현재 날짜를 입력 합니다.

```
<select id="categoryCheck" parameterType="String" resultType="dc">
        select ad_category from ad left join ps on ps_adcode=ad_code where ps_code = #{code}
 </select>
```

결제한 광고의 카테고리를 받아옵니다.

```
<select id="normalDailyCheckSelect1" parameterType="String" resultType="dc">
        select ad.ad_category,op.op_period,to_char(ps.ps_date,'YYYYMMDD') as
ps_date,to_char(ps.ps_date+op.op_period,'YYYYMMDD') as ps_date1,to_char(sysdate,'YYYYMMDD') as sd from
op left join ad on op.op_adcode=ad.ad_code left join ps on ad.ad_code=ps.ps_adcode where ps_code = #{code}
 </select>
```

카테고리와 해당 광고의 이용 기간을 받아옵니다.

```
<insert id="normalDailyCheckInsert1" parameterType="dc">
        insert into daily values(DA_SEQ.nextval,${ps_code},#{status},#{date})
 </insert>
```

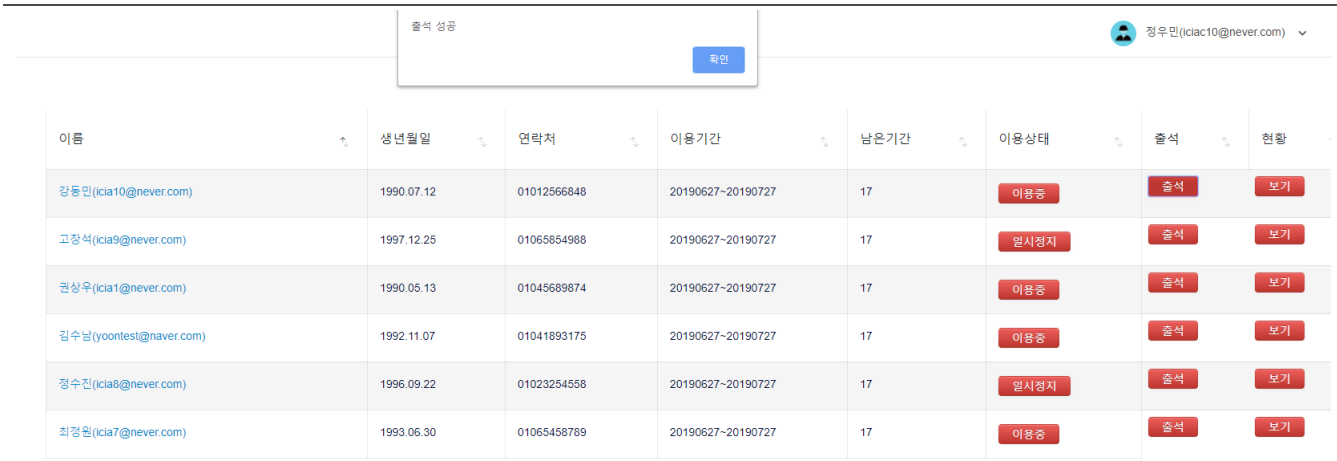
한번도 출석을 안했다면 새로운 da_code 를 생성하고 상태를 입력합니다.

## ■ FRONT-END

```
function normalCheck(id, code) {
                var id = id;
                var code = code;
                var cCode = $('#'+code).val();
                //alert('cCode = '+cCode+', code = '+code);
                if(cCode == '일시정지') {
                        alert('일시정지 상태에서는 출석이 불가능 합니다.')
                        return;
                }
                $.ajax({
                        type : "POST",
                        url : "normalcheck",
                        data : {
                                code : code
                        },
                        dataType : 'text',
                        success : function(data) {
                                alert('출석 성공');
                        },
                        error : function() {
                                alert('일반회원 입력 실패');
                        }
                });

        }
```

AJAX 를 사용하여 버튼을 클릭한 회원의 결제코드를 파라미터로 넘겨 줍니다.

## ■ 결과



## 20 JOB_NAME : 회원 이름 검색 작성자 : 권순철

■ 개요

1. 회원 목록 리스트에서 특정 회원을 이름으로 검색 할 수 있습니다.

■ CONTROLLER

```
@RequestMapping(value = "/membersearch")
public ModelAndView memberSearch(HttpServletRequest request) {
        String name = request.getParameter("name");
        String id = request.getParameter("id");
        System.out.println(name);
        System.out.println(id);

        mav = ks.getMemberSearch(name, id);

        return mav;
}
```

업체 아이디를 id 에 담고 검색한 회원 이름을 name 에 담아 getMemberSearch()에 파라미터로 넘겨 줍니다.

■ SERVICE

```
public ModelAndView getMemberSearch(String name, String id) {
        mav = new ModelAndView();

        m = new Member();

        m.setM_name(name);
        m.setM_id(id);

        List<Member> mList = null;
        mList = ksDao.getMemberSearch(m);
        System.out.println("mList = " + mList.get(0).getM_name());
        System.out.println("mList size = " + mList.size());

        if (0 != mList.size()) {
                System.out.println("member search select success");
                view = "manage/memberListC";
                mav.setViewName(view);
                mav.addObject("mList", mList);
        } else {
                System.out.println("member search select error");
                view = "redirect:memberListC.jsp";
                mav.setViewName(view);
        }

        return mav;
}
```

Member Bean 에 id 와 name 을 담고 getMemberSearch()에 m 을 파라미터로 넘겨 줍니다.
받을때는 List 타입의 Member bean 인 mList 에 담아지고 조건문에서 size 가 0 이 아니면 mav.addObject 에 mList 를 담아 memberListC.jsp 페이지로 이동합니다.

■ DAO & MAPPER

```
List<Member> getMemberSearch(Member m)
```

(1)DAO

```
<select id="getMemberSearch" parameterType="member" resultType="member">
        SELECT m_id,to_char(m_birth,'YYYY.MM.DD') as m_birth,M_NAME,M_PHONE FROM
        M WHERE M_NAME = #{m_name} AND M_ID IN(SELECT DISTINCT PS.PS_MID FROM
        PS,AD WHERE PS_ADCODE = AD.AD_CODE AND AD.AD_NAME = #{m_id})
</select>
```

업체 아이디 기준으로 회원목록에서 검색한 회원 이름을 1 개 이상 검색합니다.

■ FRONT-END

```
function memberSearch() {
                var name = $('#memsearch').val();
                //alert(name+","+sessionId);
                $.ajax({
                        type : "get",
                        url : "membersearch",
                        data : {
                                name : name,
                                id : sessionId
                        },
                        dataType : "html",
                        success : function(data) {
                                //alert(data);
                                $('#main').html(data);
                        },
                        error : function() {
                                alert('회원 검색 실패');
                        }
                });
        }
```

서버에서 받아온 데이터 data 를 #main 부분에 보여줍니다.

## ■ 결과

회원 검색 검색

| 이름 | 아이디 | 생년월일 | 연락처 | 이용상태 |
|---|---|---|---|---|
| 강동민 | icia10@never.com | 1990.07.12 | 01012566848 | 이용중 |
| 고장석 | icia9@never.com | 1997.12.25 | 01065854988 | 이용중 |
| 권상우 | icia1@never.com | 1990.05.13 | 01045689874 | 이용중 |
| 김사랑 | icia4@never.com | 1980.01.02 | 01096585542 | 이용중 |
| 김수남 | yoontest@naver.com | 1992.11.07 | 01041893175 | 이용중 |
| 김종민 | icia6@never.com | 1992.04.22 | 01062354125 | 이용중 |
| 서지원 | icia5@never.com | 1984.06.22 | 01095633214 | 이용중 |
| 윤상구 | yoontest4@never.com | 1999.01.02 | 01080765586 | 이용중 |
| 윤상기 | 18sanggi@never.com | 1993.02.27 | 01096563217 | 이용중 |
| 정민수 | icia3@never.com | 1989.05.23 | 01065448798 | 이용중 |
| 정수진 | icia8@never.com | 1996.09.22 | 01023254558 | 이용중 |

회원 검색 검색

| 이름 | 아이디 | 생년월일 | 연락처 | 이용상태 |
|---|---|---|---|---|
| 강동민 | icia10@never.com | 1990.07.12 | 01012566848 | 이용중 |

## 21 JOB_NAME : 문의 답변 등록 작성자 : 박한솔

### ■ 개요

1. 자신에게 등록된 문의를 본다
2. 그문의에 답변내용을 등록한다

### ■ CONTROLLER

(코드)

```
@ResponseBody
@RequestMapping(value = "/qanum", method = RequestMethod.POST)
public List<Question> qaNum(String qa_num) {
        List<Question> qList = new ArrayList<Question>();
        Question qau = mm.qaNum(qa_num);
        Question qa = (Question) session.getAttribute("qa");
        qList.add(qa);
        return qList;
}
```

(설명)

선택한 문의 번호를 qa_num 에 담아서 가져와 검색하여 문의 내용물을 가져온다

(코드)

```
@ResponseBody
@RequestMapping(value = "/questionreply", method = RequestMethod.POST)
public boolean questionReply(String qa_acontent,String qa_num) {
        boolean answer = mm.questionReply(qa_acontent,qa_num);
        return answer;
}
```

(설명)

작성자가 문의에 대한 답변을 저장한다

### ■ SERVICE

(코드)

```
public Question qaNum(String qa_num) {
        Question qa = mDao.qaNum(qa_num);
        session.setAttribute("qa", qa);

        return qa;
}
```

(설명)

가져온 qa_num 값을 DAO 에 qaNum 으로 넘긴고 반환된값을 Question 에 저장한다

(코드)

```
public boolean questionReply(@Param("qa_acontent") String qa_acontent, @Param("qa_num") String qa_num)
{

        boolean answer = mDao.questionReply(qa_acontent, qa_num);
        return answer;
}
```

(설명)

가져온 qa_acontent,qa_num 을 파람으로 변환후 값을 questionReply 로 넘긴후 true 또는 false 를 반환받는다

### ■ DAO & MAPPER

(코드)

```
public Question qaNum(String qa_num);
<select id="qaNum" parameterType="String" resultType="question">
        SELECT * FROM QA WHERE QA_NUM = #{qa_num}
</select>
```

(설명)

가져온 qa_num 으로 검색하여 question 에 저장한다

(코드)

```
public boolean questionReply(String qa_acontent, String qa_num);
<update id="questionReply" parameterType="hashmap">
UPDATE QA SET QA_ACONTENT = #{param1}, QA_ADATE = SYSDATE , QA_TRUE='t'
WHERE QA_NUM = #{param2}
</update>
```

(설명)

답변내용을 업데이트 해준다

## ■ FRONT-END

(코드)

```
$("#aSave").click(function() {
                var qa_num = $('#qa_num').val();
                var m_id = "${mb.m_id}";

                $.ajax({
                        url : "questionreply",
                        type : "post",
                        data : {
                                "qa_acontent" : $('#qa_acontent').val(),"qa_num" : qa_num
                        },
                        dataType : "html",
                        success : function(data) {
                                $('#qa_acontent').val("")
                                var str = "";
                                str += '<td id="aa"><a style="display: inline;" href="#myModal"
role="button" class="abtn"'
                                        +'data-toggle="modal" id="cbtn" >답변확인하기</a></td>'
                                        $('#mm').prop("style", "display: none");
                                        $('#ss').html(str);$();
                                var jsonSendAnsAlarm = JSON.stringify({qa_num:qa_num,m_id:m_id});
                                        $("#msg").val("AQ01"+jsonSendAnsAlarm);
                                        sock.send($("#msg").val());
                        },
                        error : function(error) {
                                console.log(error);
                                alert(" 실패 ");

                        }
                });

        });
```

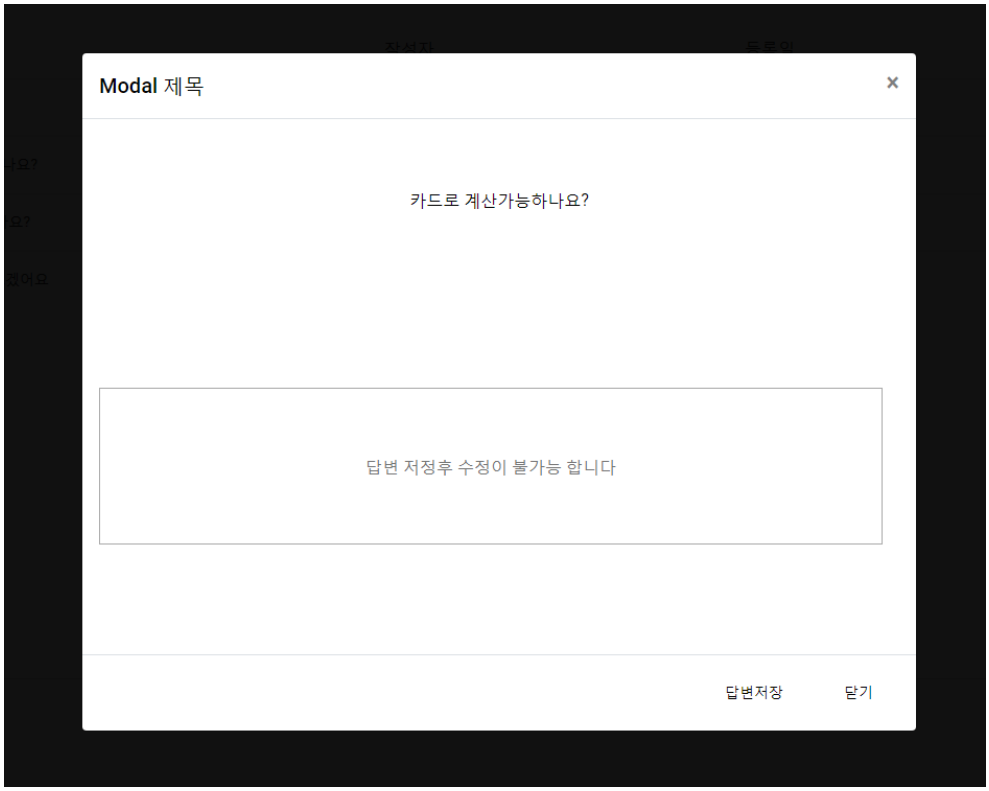
(설명)

답변내용을 qa_content, 문의번호를 qa_num 에 저장하여 ajax 통신을 한다

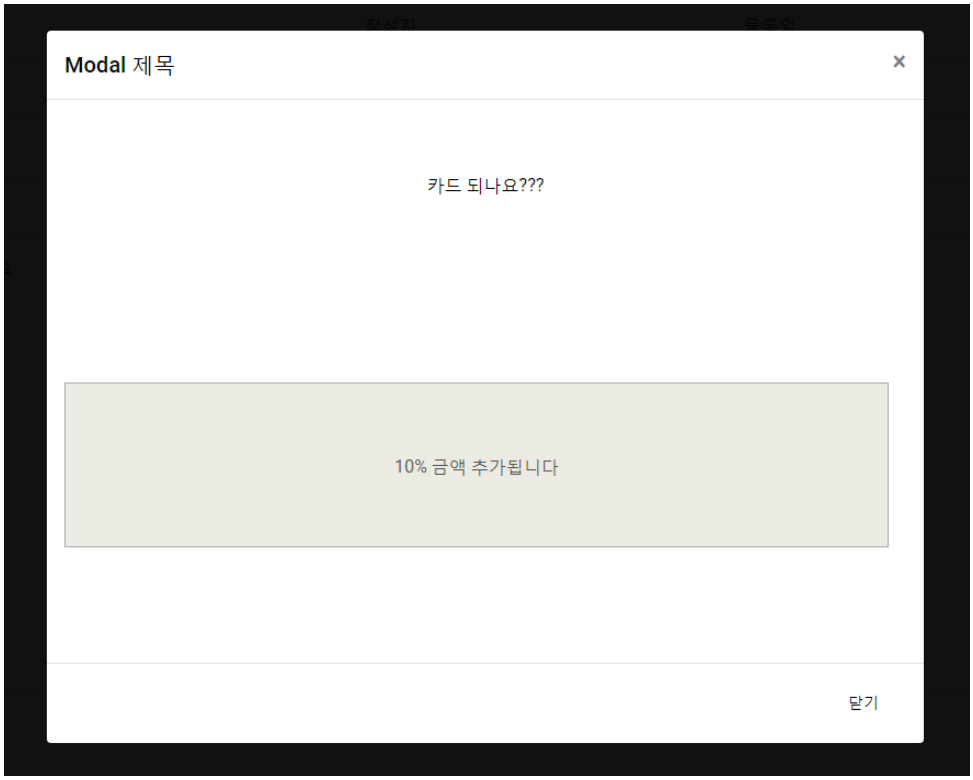
## ■ 결과

| 광고명 | 문의내용 | 작성자 | 등록일 | 답변상태 |
|---|---|---|---|---|
| 필라테스 | 카드 되나요??? | icia1@never.com | 2019-06-27 19:38:18.0 | 답변확인하기 |
| 요가 | 카드로 계산가능하나요? | icia1@never.com | 2019-07-09 15:09:41.0 | 답변하기 |
| 요가 | 좀 더 할인 가능하나요? | icia1@never.com | 2019-07-09 15:09:58.0 | 답변하기 |
| 필라테스 | 위치를 정확히 모르겠어요 | icia1@never.com | 2019-07-09 15:10:34.0 | 답변하기 |

답변할 문의나 이미 답변한 문의를 선택한다.




답변을 하지않은 경우




답변을 했을경우

문의 내용을 작성후 답변 저장을 누르면 저장이된다

문의 관리

| 광고명 | 문의내용 | 작성자 | 등록일 | 답변상태 |
|---|---|---|---|---|
| 필라테스 | 카드 되나요??? | icia1@never.com | 2019-06-27 19:38:18.0 | 답변확인하기 |
| 요가 | 카드로 계산가능하나요? | icia1@never.com | 2019-07-09 15:09:41.0 | 답변확인하기 |
| 요가 | 좀 더 할인 가능하나요? | icia1@never.com | 2019-07-09 15:09:58.0 | 답변하기 |
| 필라테스 | 위치를 정확히 모르겠어요 | icia1@never.com | 2019-07-09 15:10:34.0 | 답변하기 |

답변이 등록되면 답변 상태가 답변하기 > 답변확인하기로 바뀐다.

## 22 JOB_NAME : 소속 업체 검색 작성자 : 고정근

### ■ 개요

1. **트레이너가 검색한 업체를 보여줍니다**

### ■ CONTROLLER

```
@RequestMapping(value = "/TfindC",produces = "application/json; charset=utf8")
@ResponseBody
        public String TfindC(HttpServletRequest request) {
                String cName =request.getParameter("cName");
                String html = js.getTfindC(cName);
          return html;
        }
```

1. 검색한 업체를 cName 이란 request 에 저장합니다
2. 받은 cName 값을 가지고 js.getTfindC 서비스로 이동합니다

### ■ SERVICE

```
public String getTfindC(String name) {

                mav = new ModelAndView();
                String view = null;
                m = new Member();
                m.setC_bname(name);
                List<Member> mList = mDao.getTfindC(m);
                String html = null;
                html = makeHTMLProfileList(mList);
                return html;
        }
```

1. 받은 cName 값을 name 이란 이름으로 저장
2. Member bean m 에 값을 저장
3. Dao 에서 리턴 받은 값을 mList 에 저장하여 리턴

### ■ DAO & MAPPER

**Dao**

```
public List<Member> getTfindC(Member m);
```

**Mapper**

```
<select id="getTfindC" parameterType="member" resultType="member">
        <![CDATA[
                select c.c_bname, m.m_addr, m.m_id from m inner join c on m.m_id=c.c_id where c_bname
           like '%${c_bname}%'
                ]]>
</select>
```

1. 회원테이블과 업체테이블을 조인
2. 검색할 업체의 업체명, 주소를 검색하여 저장

### ■ FRONT-END

**jsp**

```
<div class="form-group">
<div class="input-group">
<input type="text" name="c_bname" class="form-control form-control-lg border-left-0"
id="sample6_address" placeholder="현재 소속업체 ID: ${m.t_cid}"><button type="button" id =
"aa"class="btn btn-outline-secondary btn-md">업체 검색</button>
</div>
</div>
<div id="test">
```

**Script**

```
        $.ajax({
                type : "get",
                url : "TfindC",
                data : {
                        cName : cName
                        },
                dataType : "html",
                success : function(data) {
                        $("#test").html(data);
                error : function(er) {
                        alert('업체 검색 실패');
                        }
                });
});
```

1. 프로필 페이지 전환 시 소속 페이지가 있으면 소속업체를 업체 검색 란에 placeholder 로 노출
2. 업체 검색시 ajax 를 통해 id 가 test 인 div 에 검색한 값을 노출
3. 웹소켓으로 해당업체의 알림에 요청한 목록을 노출한다

■ **결과**

## 23 JOB_NAME : 업체 소속 요청 작성자 : 고정근

### ■ 개요

**검색한 업체를 소속요청하여 해당 업체에 알림 기능 구형**

### ■ CONTROLLER

```
@RequestMapping(value = "/acceptrequest",produces = "application/json; charset=utf8")
        @ResponseBody
        public String acceptrequest(String c_id) {
                Member mb = (Member) session.getAttribute("mb");
                String text = js.acceptrequest(mb,c_id);
                return text;
        }
```

1.해당 트레이너의 아이디를 c_id 에 값을 담고 member bean mb 에 저장

### ■ SERVICE

```
public String acceptrequest(Member mb, String c_id) {
                String text = null;
                m = new Member();
                m.setM_id(mb.getM_id());
                m.setC_id(c_id);
                int count = mDao.comfirm(m);
                try {
                        if (count == 1)
                                mDao.acceptrequestupdate(m);
                        text = "update";
                        if (count == 0)
                                mDao.acceptrequestInsert(m);
                        text = "insert";
                } catch (Exception e) {
                }
                return text;
        }
```

1. **받아온 값을 comfirm 에 넣고 소속여부 확인**
2. **만약 이전에 요청한 업체가 있다면 DB 에 UPDATE**
   **요청한 업체가 없다면 DB 에 INSERT**
3. **반환 받은 값을 text 에 저장하여 return**

### ■ DAO & MAPPER

**Dao**

```
public boolean acceptrequestInsert(Member m);
```

**Mapper**

```
<insert id="acceptrequestInsert"  >
        INSERT INTO YESORNO VALUES(#{c_id},#{m_id},'진행',SYSDATE)
        </insert>
<update id="acceptrequestupdate" parameterType="member">
      UPDATE YESORNO SET YN_STATE= '진행',YN_COMPANY=#{c_id} WHERE YN_TRAINER= #{m_id}
       </update>
```

1. 소속업체가 있을때는 소속상태를 '진행'으로 update
2. 소속업체가 없을때는 소속상태를 '진행'으로 insert

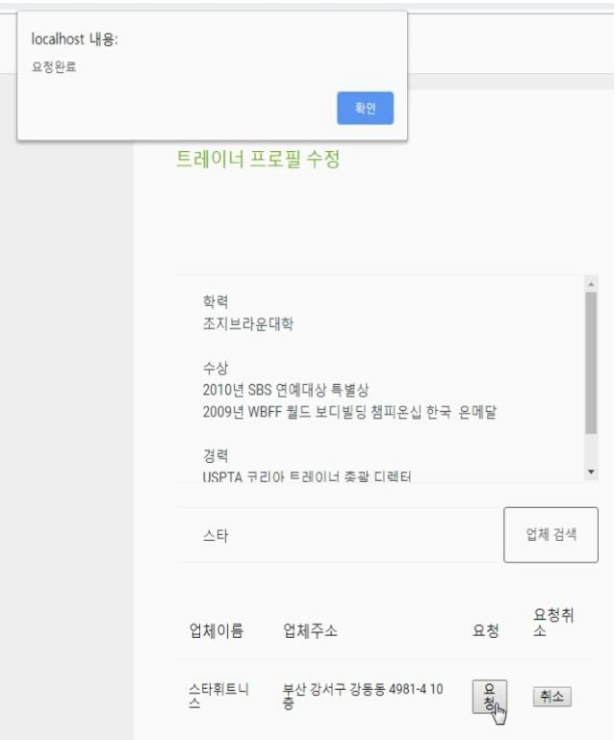
### ■ FRONT-END

```
<div class="form-group">
        <div class="input-group">
            <input type="text" name="c_bname" class="form-control form-control-lg border-left-0"
            id="sample6_address" placeholder="현재 소속업체 ID: ${m.t_cid}">
```

```
            <button type="button" id = "aa" class="btn btn-outline-secondary btn-md">업체 검색</button>
        </div>
    </div>
    <div id="test">
```

1. 검색된 업체에 요청 버튼 클릭
2. 요청 완료

## ■ 결과



## 24 JOB_NAME : 소속 요청 취소 작성자 : 고정근

### ■ 개요

**요청한 업체를 요청 취소**

### ■ CONTROLLER

```
@RequestMapping(value = "/cancel",produces = "application/json; charset=utf8")
        public @ResponseBody String cancel(HttpServletRequest request) {
                Member mb = (Member) session.getAttribute("mb");
                String html = js.cancel(mb);
                return html;
        }
```

1. Session 에 있는 mb 값을 가져와 js.cancel 서비스로 넘겨준다

### ■ SERVICE

```
public String cancel(Member mb) {
                String cc = null;
                String id = mb.getM_id();
                if (mDao.cancel(id)) {
                        System.out.println("취소완료");
                        cc = "취소";
                }
                else {
                        System.out.println("취소실패");
                }
                return cc;
        }
```

1. Mb 에 있는 id 값을 추출 한 후 mDao.cancel 에 같이 넘긴다
2. DB 에 값이 UPDATE 되면 취소 완료 / UPDATE 가 되지않으면 취소 실패

### ■ DAO & MAPPER

**DAO**

```
public boolean cancel(String id);
```

**Mapper**

```
<update id="cancel" parameterType="String" >
                UPDATE YESORNO SET YN_STATE= '취소' WHERE YN_TRAINER = #{id}
        </update>
```

1. Yesorno 테이블에 가져온 아이디 값의 상태를 '취소'로 변경한다

### ■ FRONT-END

```
<div class="form-group">
        <div class="input-group">
            <input type="text" name="c_bname" class="form-control form-control-lg border-left-0"
            id="sample6_address" placeholder="현재 소속업체 ID: ${m.t_cid}">
            <button type="button" id = "aa" class="btn btn-outline-secondary btn-md">업체 검색</button>
        </div>
</div>
<div id="test">
```

1. 검색된 업체에 취소 버튼 클릭 취소 완료

## 25 JOB_NAME : 프로필 수정 (트레이너) 작성자 : 고정근

### ■ 개요

**트레이너의 프로필 수정**

### ■ CONTROLLER

```
public ModelAndView profileComplete(HttpServletRequest request) {
                    Member mb = (Member) session.getAttribute("mb");
                    String t_career=request.getParameter("t_career");
                    mav = js.profileComplete(mb,t_career);
                    return mav;
          }
```

1. 해당 ID 를 사용하기 위해 Session에 있는 mb 호출
2. 수정된 커리어의 수정본을 t_career에 저장

### ■ SERVICE

```
public ModelAndView profileComplete(Member mb, String t_career) {
                    mav = new ModelAndView();
                    mb.setT_career(t_career);
                    mDao.profileComplete(mb);
                    String view = "manage/trainer/trainerMain";
                    mav.setViewName(view);
                    return mav;
          }
```

1. **수정된 t_career 값을 mb 에 저장**
2. **mDao 의 profileComplete 기능 실행**
3. **완료시 화면전환은 트레이너 메인페이지로 이동**

### ■ DAO & MAPPER

**DAO**

```
public void profileComplete(Member mb);
```

**Mapper**

```
<update id="profileComplete" parameterType="member">
                    UPDATE TRAINER SET T_CAREER=#{t_career} WHERE T_ID= #{m_id}
          </update>
```

1. 해당 id 의 값의 t_career 를 수정된 t_career 로 업데이트
2. 수정된 t_career 를 m 에 저장

### ■ FRONT-END

```
<div class="form-group">
          <div class="input-group">
                    <textarea rows="8" cols="8" name="t_career" id="t_career"
                class="form-control form-control-lg border-left-0">
              ${m.t_career}</textarea>
            </div>
          </div>
```

1. 저장된 t_career 를 m.t_career 로 호출

### ■ 결과

트레이너 프로필 수정
학력
조지브라운대학
수상
2010년 SBS 연예대상 특별상
2009년 WBFF 월드 보디빌딩 챔피온십 한국 은메달
경력
USPTA 코리아 트레이너 총괄 디렉터

## 26 JOB_NAME : 트레이너 개인정보 수정 작성자 : 박한솔

■ 개요

**1. 트레이너 개인 정보를 가져온다**

■ CONTROLLER

(코드)

```
@ResponseBody
        @RequestMapping(value = "/trainerModifyT" ,method = RequestMethod.GET)
        public ModelAndView trainerModifyT() {
                Member mb = (Member) session.getAttribute("mb");
                String m_id = mb.getM_id();
                mav = mm.trainerModifyT(m_id);
                return mav;
        }
```

(설명)

로그인시 저장한 세션의 값중 아이디 값을 넘겨준다

■ SERVICE

(코드)

```
        public ModelAndView trainerModifyT(String m_id) {
                mav = new ModelAndView();
                m = new Member();
                m = mDao.trainerModifyT(m_id);
                String pf_image = mDao.pfimage(m_id);
                session.setAttribute("mb", m);
                mav.addObject("m_id", m.getM_id());
                mav.addObject("m_name", m.getM_name());
                mav.addObject("m_phone", m.getM_phone());
                mav.addObject("m_birth", m.getM_birth());
                mav.addObject("m_addr", m.getM_addr());
                mav.addObject("m_exaddr", m.getM_exaddr());
                mav.addObject("m_kind", m.getM_kind());
                mav.addObject("pf_image", m.getPf_image());
                view = "manage/infoModifyT";
                mav.setViewName(view);

                return mav;
        }
```

(설명)

가져온 아디디값으로 검색 EL 로 출력할수있게 addObjet 에 저장 한후 반환 해준다

■ DAO & MAPPER

(코드)

```
public Member trainerModifyT(String m_id);
public String pfimage(String m_id);

        <select id="trainerModifyT" parameterType="String"
                resultType="member">
                SELECT * from t_modify
                WHERE m_id = #{m_id}
        </select>
        <select id="pfimage" parameterType="String" resultType="String">
                SELECT *
                FROM PF WHERE PF_ID=#{m_id}
        </select>
```

■ 결과

트레이너 내 정보 수정

프로필사진

파일 선택 다운로드.jpg

iciat10@never.com

패스워드를 입력해주세요

패스워드 확인

박두식

01094859513

1998/07/08

서울 강남구 도산대로 404 퍼스트 빌딩 3층 피티숍

주소 검색

퍼스트 빌딩 3층 피티숍

변경사항저장

정보를가져온다

## 27 JOB_NAME : 일반프로그램 이용 상태 전환 작성자 : 권순철

■ 개요

1. 일반 피트니스(헬스장)을 이용하는 회원은 기간이 남아 있는 동안에는 일시정지 기능을 이용 할 수 있습니다.

■ CONTROLLER

```
@RequestMapping(value = "/changestate",produces = "application/text; charset=utf8")
public @ResponseBody String changeState(HttpServletRequest request) {

        String gson = ks.changeState(request);

        return gson;
}
```

request 에는 일시정지 또는 이용중으로 변경하는 회원의 결제 코드와 변경할 이용 상태가 담겨져 changeState()에 파라미터로 넘겨집니다.

■ SERVICE

```
public String changeState(HttpServletRequest request) {
        Map<String, String> map = new HashMap<String, String>();
        Gson gson = new Gson();
        String str = null;

        String code = request.getParameter("code");
        String status = request.getParameter("status");
        map.put("code", code);
        map.put("status", status);

        try {
                ksDao.changeState(map);
                // ksDao.trainerDisconDelete(map);

                System.out.println("trainerDiscon success");
        } catch (Exception e) {
                System.out.println("trainerDiscon fail");
        }

        return "success";

}
```

**Map 타입의 map 변수를 생성하여 받아온 code 와 status 를 담습니다.**
**changeState()에 map 을 파라미터로 넘겨 줍니다.**

■ DAO & MAPPER

```
void changeState(Map<String, String> map)
```

(1)DAO

```
<update id="changeState" parameterType="HashMap">
        update daily set da_status = #{status} where da_pscode = #{code}
</update>
```

(2)MAPPER

Daily 테이블의 da_status 를 받아온 status 상태로 변경 합니다.

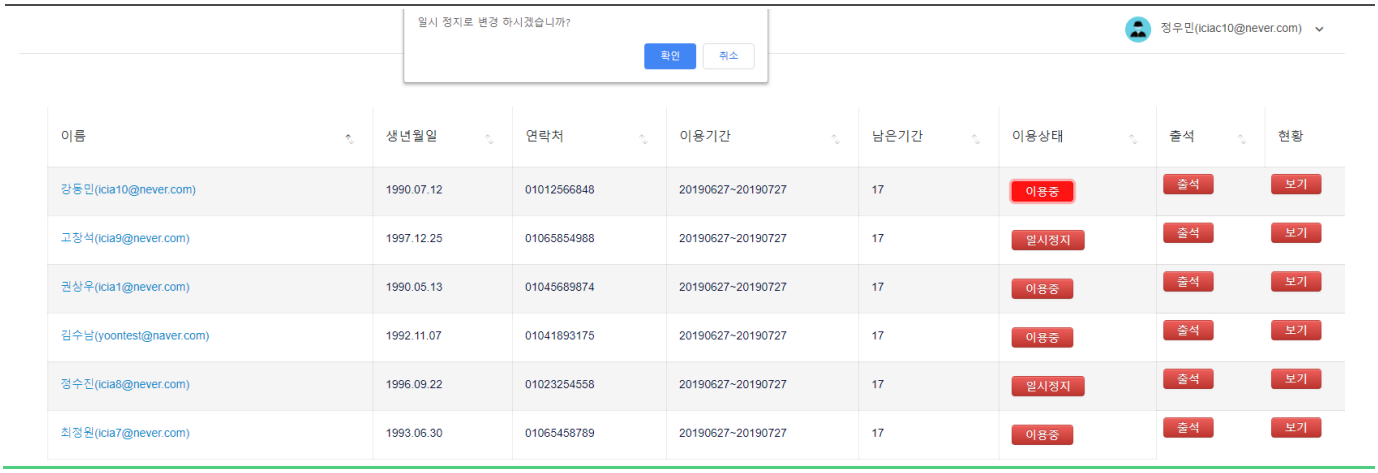
■ FRONT-END

```
function statusChange(code1,status1) {
        var code = code1;
        var status = status1;
        //alert(code+","+status);
        var start = '이용중으로 변경 하시겠습니까?';
        var stop = '일시 정지로 변경 하시겠습니까?';
        var str = '';
```

```
        if(status =='이용중'){
                str = stop;
                status = '일시정지';
        }else if(status == '일시정지') {
                str = start;
                status = '이용중';
        }

        if (confirm(str) == true) {
                $.ajax({
                        type : 'post',
                        url : 'changestate',
                        data : {
                                code : code,
                                status : status
                        },
                        dateType : 'json',
                        success : function(data) {

                                $('#'+code).html(status);
                                $('#'+code).val(status);
                        },
                        error : function(error) {
                                alert("이용상태 변경 실패");
                        }
                });
        } else {
                return;
        }
}
```

AJAX 를 실행전에 현재 상태를 체크한 후 상태에따라 status 변수에 이용중,일시정지를 담습니다.
Code 와 Status 를 changestate URL 로 넘깁니다.

## ■ 결과



## 28 JOB_NAME : 트레이너 요청에 대한 수락/거절 작성자 : 권순철

■ 개요

1. 개인 트레이너가 업체에 지원(요청)을 하게 되면 업체는 수락,거절을 할 수 있습니다.

■ CONTROLLER

```
@RequestMapping(value = "/trainerjoin",produces = "application/text; charset=utf8")
public @ResponseBody String trainerJoin(HttpServletRequest request) {
        String gson = ks.trainerJoin(request);
        return gson;
 }
```

■ SERVICE

```
public String trainerJoin(HttpServletRequest request) {
Map<String, String> map = new HashMap<String, String>();
ArrayList<HashMap<String, String>> hMap = new ArrayList<HashMap<String, String>>();
Gson gson = new Gson();
String str = null;
String state = request.getParameter("state");

if (state.equals("1")) {
        state = "수락";
} else if (state.equals("0")) {
        state = "거절";
}

String cid = request.getParameter("cid");
String id = cid;
String tid = request.getParameter("tid");
map.put("state", state);
map.put("cid", cid);
map.put("tid", tid);

try {
        if (state.equals("수락")) {
                ksDao.trainerJoinUpdate(map);
                ksDao.trainerJoinInsert(map);

                hMap = ksDao.getTrainerJoinList(id);
        } else if (state.equals("거절")) {
                ksDao.trainerJoinUpdate(map);

                hMap = ksDao.getTrainerJoinList(id);
        }

        str = gson.toJson(hMap);
        System.out.println(str);
        System.out.println("trainerJoin update,insert success");
} catch (Exception e) {
        System.out.println("trainerJoin update,insert fail");
}

return str;
}
```

■ DAO & MAPPER

**(1) DAO**

```
void trainerJoinUpdate(Map<String, String> map);
void trainerJoinInsert(Map<String, String> map);
```

trainerJoinUpdate 를 실행하여 요청을 수락으로 변경합니다.

trainerJoinInsert 를 실행하여 트레이너 회원에 업체 아이디를 입력 합니다.

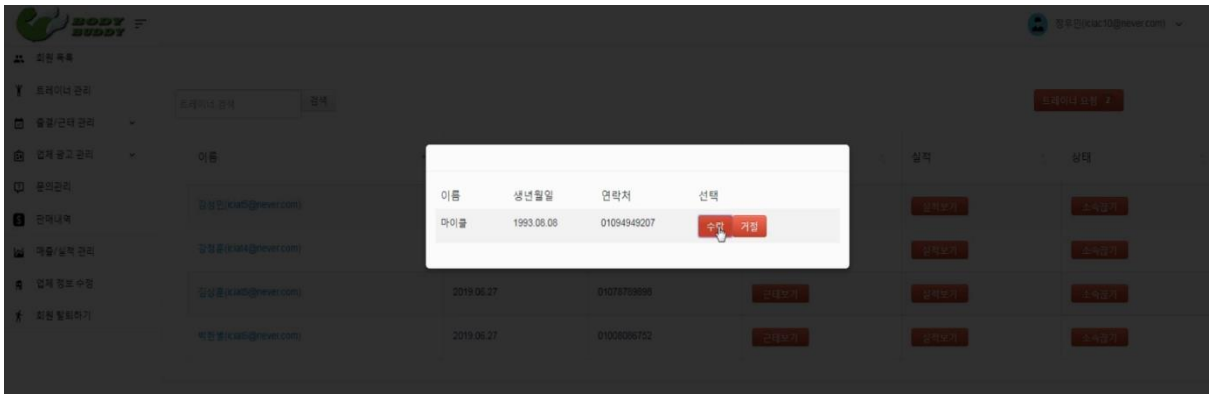
**(2) MAPPER**

```
<update id="trainerJoinUpdate" parameterType="HashMap">
    update yesorno set yn_state = #{state}, yn_date = sysdate where yn_trainer = #{tid} and yn_company= #{cid}
</update>
<update id="trainerJoinInsert" parameterType="HashMap">
         update trainer set t_cid = #{cid} where t_id = #{tid}
</update>
```

■ FRONT-END

```
function trainerJoin(state, tid) {
var state = state;
var tid = tid;
var ttid = "#" + tid;
         $.ajax({
                  type : 'post',
                  url : 'trainerjoin',
                  data : {
                           state : state,
                           cid : sessionId,
                           tid : tid
                  },
                  dateType : 'json',
                  success : function(data) {
                           console.log(data);
                           $('#modal').modal('hide');
                                    //////////소켓 알림
                           if (state == 1) {
                                    var jsonSendSosokAcptAlarm = JSON.stringify({
                                             c_id : sessionId,
                                             t_id : tid
                                    });
                                    $("#msg").val("SS02" + jsonSendSosokAcptAlarm);
                                             console.log("WebSock7", sock);
                                             console.log($("#msg").val());
                                             sock.send($("#msg").val());
                                    } else if (state == 0) {//////////소속거절 웹소켓 알림
                                             var jsonSendSosokRjctAlarm = JSON.stringify({
                                                      c_id : sessionId,
                                                      t_id : tid
                                    });
                                    $("#msg").val("SS03" + jsonSendSosokRjctAlarm);
                                             console.log("WebSock8", sock);
                                             console.log($("#msg").val());
                                             sock.send($("#msg").val());
                                    }
                                    /////////////
                           },
                           error : function(error) {
                                    alert("트레이너 수락 거절 실패")
                           }
                  });
         }
```

■ 결과




업체가 요청 수락을 하게 되면 트레이너 관리 페이지에 해당 트레이너가 추가가 되고, 거절하게 되면 요청 목록에서 없어지며 해당 테이블에 요청에서 거절로 바뀌게 됩니다.

## 29 JOB_NAME : 트레이너 소속 끊기 작성자 : 권순철

■ 개요

1. 소속된 트레이너를 제외 시키고 싶을 때 소속 끊기를 누르면 트레이너 리스트에서 삭제 됩니다.

■ CONTROLLER

```
@RequestMapping(value = "/trainerdiscon",produces = "application/text; charset=utf8")
public @ResponseBody String trainerDiscon(HttpServletRequest request) {
        String gson = ks.trainerDiscon(request);
        return gson;
        }
```

request 에는 트레이너 아이디와 업체 아이디가 담겨있어 trainerDiscon()의 파라미터로 넘깁니다.

■ SERVICE

```
public String trainerDiscon(HttpServletRequest request) {
        Map<String, String> map = new HashMap<String, String>();
        Gson gson = new Gson();
        String str = null;

        String cid = request.getParameter("cid");
        String id = cid;
        String tid = request.getParameter("tid");
        map.put("cid", cid);
        map.put("tid", tid);

        try {
                ksDao.trainerDiscon(map);
                ksDao.trainerDisconDelete(map);
                System.out.println("trainerDiscon success");
        } catch (Exception e) {
                System.out.println("trainerDiscon fail");
        }

        return "success";
}
```

**Map(String,String)타입의 map 변수를 생성하고 트레이너 아이디와 업체 아이디를 담아 trainerDiscon(),trainerDisconDelete()의 파라미터로 map 을 넘깁니다.**

■ DAO & MAPPER

```
void trainerDiscon(Map<String, String> map)
void trainerDisconDelete(Map<String, String> map)
```

**(1)DAO**

---

```
<update id="trainerDiscon" parameterType="HashMap">
        update trainer set t_cid = '소속업체없음' where t_id =#{tid} and t_cid = #{cid}
</update>
<delete id="trainerDisconDelete" parameterType="HashMap">
        delete from yesorno where yn_company = #{cid} and yn_trainer = #{tid}
</delete>
```

**(2)MAPPER**

trainerDiscon 은 트레이너 테이블의 소속 컬럼 내용을 '소속업체없음'으로 변경 합니다.

trainerDisconDelete 는 요청,수락 테이블에 해당 트레이너와 업체 아디디 기준으로 찾아서 튜플을 삭제 합니다.

---

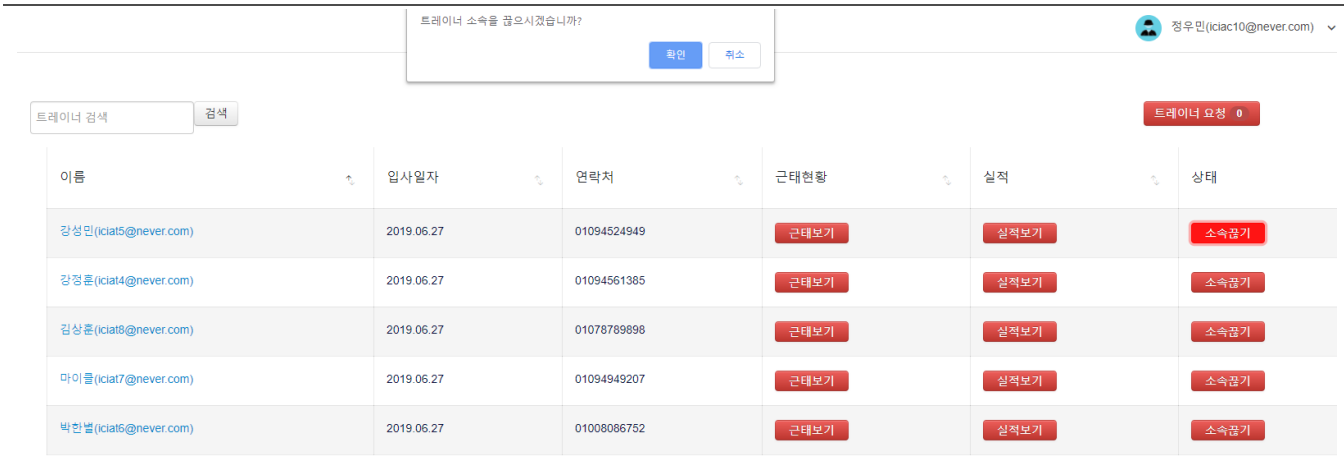
■ FRONT-END

```
function delete_event(tid) {
        if (confirm("트레이너 소속을 끊으시겠습니까?") == true) {
                var tid = tid;
                //alert(tid);
```

```
                $.ajax({
                        type : 'post',
                        url : 'trainerdiscon',
                        data : {
                                cid : sessionId,
                                tid : tid
                        },
                        dateType : 'json',
                        success : function(data) {
                                alert(data);

                                $('#modal').modal('hide');
                        },
                        error : function(error) {
                                alert("트레이너 소속끊기 실패")
                        }
                });
        } else {
                return;
        }
}
```

소속 끊기를 하기전에 한번 더 물어 확인 버튼을 누르면 AJAX 부분이 실행이 됩니다.

## ■ 결과



## 30 JOB_NAME : 소속 트레이너 근태 상태 입력 작성자 : 권순철

■ 개요

1. 소속된 트레이너의 근태를 관리하기 위해서 출근,휴가,결근으로 근태를 입력합니다.

■ CONTROLLER

```
@RequestMapping(value = "/dailycheckinsert")
public ModelAndView dailyCheckInsert(HttpServletRequest request) {
        System.out.println(request.getParameter("tid"));
        System.out.println(request.getParameter("cid"));
        System.out.println(request.getParameter("status"));

        mav = ks.dailyCheckInsert(request);

        return mav;
}
```

request 안에는 업체 아이디,트레이너 아이디,근태 상태 담겨 있어 dailyCheckInsert 의 파라미터로 넘겨줍니다.

■ SERVICE

```
public ModelAndView dailyCheckInsert(HttpServletRequest request) {
        mav = new ModelAndView();

        m = new Member();
        m.setDt_tid(request.getParameter("tid"));
        m.setDt_cid(request.getParameter("cid"));
        m.setDt_status(request.getParameter("status"));

        boolean dci = false;

        try {
                dci = ksDao.trainerDailyCheckInsert(m);
                mav.addObject("msg", "insert");
        } catch (Exception e) {
                dci = ksDao.trainerDailyCheckUpdate(m);
                mav.addObject("msg", "update");
        }

        if (dci) {
                System.out.println("dailyCheckInsert success");
                view = "manage/company/trainerDailyCheck";
                mav.setViewName(view);

        } else {
                System.out.println("dailyCheckInsert fail");
                view = "manage/company/trainerDailyCheck";
                mav.setViewName(view);
        }

        return mav;
}
```

Member m 을 생성하여 받아온 업체 아이디,트레이너 아이디, 근태 상태를 담습니다.

trainerDailyCheckInsert()에 파라미터로 m 을 넘겨주고 만약 입력이 실패 됐다면 당일에 입력이 됐다고 보고 ectch 에서 다시 업데이트를 실행합니다.

■ DAO & MAPPER

```
boolean trainerDailyCheckInsert(Member m)
boolean trainerDailyCheckUpdate(Member m)
```

**파라미터로 Member Bean 타입의 m 를 파라미터로 넘겨줍니다.**

**(1) DAO**

```
<insert id="trainerDailyCheckInsert" parameterType="member">
        INSERT INTO DAILYCHECKT VALUES(#{dt_tid},#{dt_cid},TO_CHAR(sysdate,'YYYYMMDD'),#{dt_status})
</insert>
```

```
<update id="trainerDailyCheckUpdate" parameterType="member">
        Update dailycheckt set dt_status = #{dt_status} where dt_tid = #{dt_tid} and
        dt_cid = #{dt_cid} and dt_date = to_char(sysdate,'YYYYMMDD')
 </update>
```

*trainerDailyCheckInsert* 는 현재 시스템의 년,월,일과 트레이너와 업체의 아이디를 입력합니다.

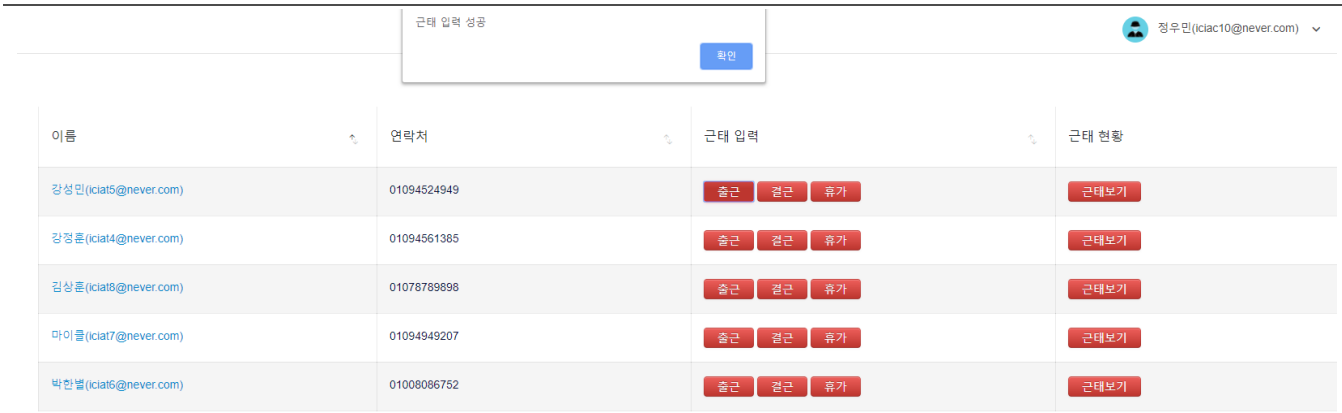
*trainerDailyCheckUpdate* 는 해당 당일에 입력 된게 있다면 업데이트를 통해서 dt_status 의 상태를 변경 합니다

## ■ FRONT-END

```
function dailyCheck(status, id, cid) {
        var status = status;
        var tid = id;
        var cid = cid
        $.ajax({
                type : "POST",
                url : "dailycheckinsert",
                data : {
                        tid : tid,
                        status : status,
                        cid : cid
                },
                dataType : 'html',
                success : function(data) {
                        alert("근태 입력 성공");
                },
                error : function() {
                        alert('트레이너 근태 입력 실패');
                }
        });
}
```

출석 버튼을 클릭하면 dailyCheck 가 실행되며 파라미터로 업체 아이디, 트레이너 아이디, 근태 버튼 상태를 넘겨줍니다.

## ■ 결과



## 31 JOB_NAME : 정보 수정 페이지 이동 작성자 : 권순철

■ 개요

1. 업체 회원의 기본 입력 정보를 불러온다.

■ CONTROLLER

```
@RequestMapping(value = "/infomodifyc")
        public ModelAndView infoModifyC(HttpServletRequest request) {
                String id = request.getParameter("id");
                mav = ks.getInfomodifyC(id);

                return mav;
        }
```

1. 업체 회원 아이디(id)를 파라미터로 받음
2. kwonService 의 getInfomodifyC(id)실행

■ SERVICE

```
public ModelAndView getInfomodifyC(String id) {
                mav = new ModelAndView();
                com = new Company();
                m = new Member();

                System.out.println("getInfomodifyC mDao in");
                com = ksDao.getInfomodifyC(id);
                m = ksDao.getinfoModifyImage(id);

                if (com != null) {
                        System.out.println("getInfomodifyC success");
                        view = "manage/infoModifyC";
                        mav.setViewName(view);
                        mav.addObject("com", com);
                        mav.addObject("m", m);
                } else {
                        System.out.println("getInfomodifyC error");
                }

                return mav;
        }
```

1. Company Bean 에 ksDao.getInfomodifyC(id)의 실행 결과 값 업체 회원 정보를 담는다.
2. Member Bean 에 ksDao.getinfomodifyImage(id)의 결과 값 이미지 파일명을 담는다.
3. infoModifyC.jsp 으로 이동

■ DAO & MAPPER

```
Company getInfomodifyC(String id);
Member getinfoModifyImage(String id);
```

**(1) DAO**

mapper 에 있는 getInfomodifyC, getinfomodifyImage 로 파라미터 id 를 보냄

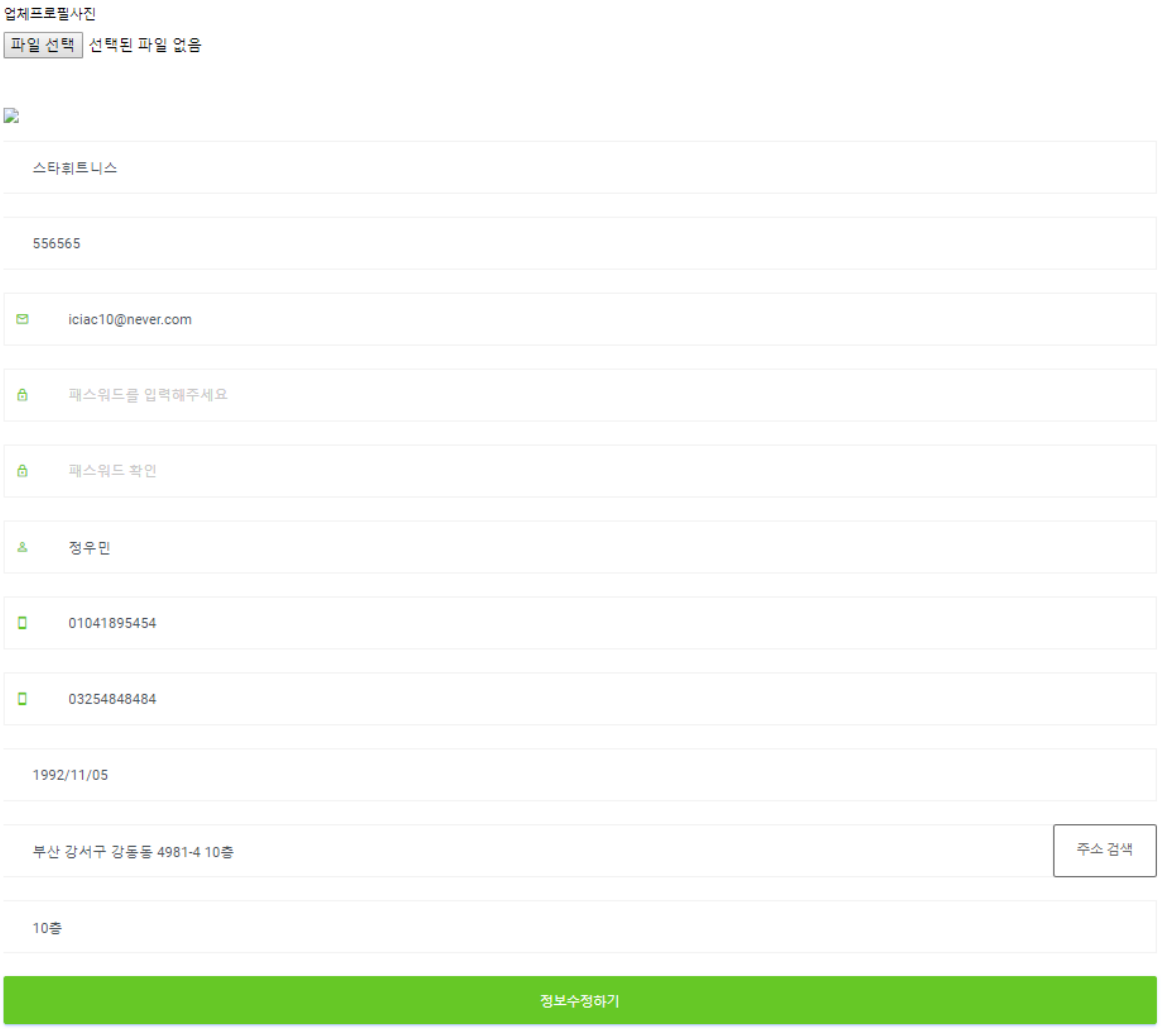
---

```
<select id="getInfomodifyC" resultType="company">
    Select c_id,c_num,c_bname,c_bphone,m_id,m_pw,m_name,m_phone,to_char(m_birth,'YYYY/MM/DD')
as m_birth,m_addr,m_exaddr,m_kind from c left join m on m_id = c_id where m_id = #{id}
</select>
<select id="getinfoModifyImage" parameterType="String" resultType="member">
        select pf_image from profilephoto where pf_id = #{id}
</select>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 업체 회원 정보와 이미지 파일을 SELECT 한다.

■ 결과




**자신의 개인 정보를 보여 줍니다.**

## 32 JOB_NAME : 프로필 수정 완료 작성자 : 권순철

■ 개요

1. 자신의 프로필 사진, 전화번호, 주소 등을 수정하여 완료한다.

■ CONTROLLER

```
@ResponseBody
@RequestMapping(value = "/infomodifyupdate" ,method = RequestMethod.POST)
public String infoModifyUpdate(MultipartHttpServletRequest multi) {
        String gson = ks.infoModifyUpdate(multi);
        System.out.println(gson);

        return gson;
}
```

1. MultipartHttpServletRequest 형식의 multi 파라미터로 회원정보 페이지의 정보를 받음
2. kwonService 의 `infoModifyUpdate(multi)` 실행

■ SERVICE

```
public String infoModifyUpdate(MultipartHttpServletRequest multi) {
Map<String, String> fMap = new HashMap<String, String>();
BCryptPasswordEncoder pwdEncoder = new BCryptPasswordEncoder();
mav = new ModelAndView();
com = new Company();
m = new Member();

String root = multi.getSession().getServletContext().getRealPath("/");
System.out.println("root=" + root);
String path = root + "resources/upload/";
File dir = new File(path);
if (!dir.isDirectory()) { // upload폴더 없다면
        dir.mkdirs(); // upload폴더 생성
}

String m_id = multi.getParameter("c_id");
System.out.println("m_id 확인" + m_id);
String m_phone = multi.getParameter("m_phone");
System.out.println("m_phone 확인" + m_phone);
String m_addr = multi.getParameter("m_addr");
System.out.println("m_addr 확인" + m_addr);
String m_exaddr = multi.getParameter("m_exaddr");
System.out.println("m_exaddr 확인" + m_exaddr);
String c_bphone = multi.getParameter("c_bphone");
System.out.println("c_bphone 확인" + c_bphone);
String m_pw = multi.getParameter("m_pw");
System.out.println("m_pw 확인" + m_pw);
String fileCheck = multi.getParameter("fileCheck").trim();
System.out.println("fileCheck 확인" + fileCheck);
int checkNum = Integer.parseInt(fileCheck);
System.out.println("checkNum = " + checkNum);

com.setM_id(m_id);
com.setM_phone(m_phone);
com.setM_addr(m_addr);
com.setM_exaddr(m_exaddr);
com.setC_bphone(c_bphone);

try {
        if (m_pw != null) {
                System.out.println("pwd null이 아니면 실행");
                com.setM_pw(pwdEncoder.encode(m_pw));
                System.out.println("pwd update start");
```

```
                    ksDao.infoModifyPwUpdate(com);
                    System.out.println("pwd update success");
            }
    } catch (Exception e) {
            System.out.println("pwd update fail");
    }

    MultipartFile file = multi.getFile("pf_image");
    String oriFileName = null;
    String sysFileName = null;
    int cNum = 0;

    try {
            if (checkNum >= 1) {
                    System.out.println("checkNum 파일 있음" + "," + checkNum);
                    oriFileName = file.getOriginalFilename();
                    sysFileName = System.currentTimeMillis() + "."
                                                    + oriFileName.substring(oriFileName.lastIndexOf(".") +
    1);
                    System.out.println("oriFileName=" + oriFileName);
                    System.out.println("sysFileName=" + sysFileName);
                    fMap.put("pf_id", m_id);
                    fMap.put("pf_image", sysFileName);
                    System.out.println("sysfilename = " + sysFileName);

            }
            if (sysFileName != null) {
                    file.transferTo(new File(path + sysFileName));

                    cNum = ksDao.profilePhotoSelect(m_id);
                    if (cNum >= 1) {
                            ksDao.fileUpdate(fMap);
                    } else if (cNum == 0) {
                            ksDao.fileInsert(fMap);
                    }

            }
    } catch (Exception e) {
            System.out.println("file Insert,Update fail");
    }

    try {
            System.out.println("infoModifyUpdate start");
            ksDao.infoModifyMemberUpdate(com);
            ksDao.infoModifyCompanyUpdate(com);
            System.out.println("infoModifyUpdate success");
    } catch (Exception e) {
            System.out.println("infoModifyUpdate fail");
    }

    System.out.println("m_phone+c_bphone+m_addr+m_exaddr=" + m_phone + c_bphone + m_addr + m_exaddr +
    com.getM_pw() + m_id);
    Gson gson = new Gson();
    String ajGson = null;
    try {
            com = ksDao.getInfomodifyC(m_id);
            m = ksDao.getinfoModifyImage(m_id);
            ajGson = gson.toJson(com) + gson.toJson(m);
           mav.setViewName("forward:infomodifyc");

    } catch (Exception e) {
            System.out.println("정보수정 실패");
    }

            return ajGson;
    }
```

1. Company Bean 에 받은 데이터를 담는다.
2. 패스워드를 입력 받았는지 검사하여 받았다면 암호화 하여 Company Bean 에 담는다.
3. infoModifyPwUdate(com)으로 패스워드를 전달한다.

4. MultipartFile 을 이용하여 받아온 파일을 담는다.
5. checkNum 으로 0 이 아니면 파일명을 변경하여 fMap 에 아이디와 파일을 담는다.
6. 기존의 파일이 있다면 fileUpdate(fMap)을 실행하고 없다면 fileInsert(fMap)을 실행한다.
7. infoModifyMemberUpdate(com),infoModifyCompanyUpdate(com)를 실행한다.
8. getInfomodifyC(m_id),getinfoModifyImage(m_id)를 실행하여 각각 Company,Member Bean 에 담는다.
9. Company,Member Bean 에 담은 값을 json 형식으로 변환한다.

## ■ DAO & MAPPER

```
void infoModifyPwUpdate(Company com);
int profilePhotoSelect(String m_id);
void fileUpdate(Map<String, String> fMap);
void fileInsert(Map<String, String> fMap);
void infoModifyMemberUpdate(Company com);
void infoModifyCompanyUpdate(Company com);
```

**(1) DAO**

```
<update id="infoModifyPwUpdate" parameterType="company">
    update member set m_pw = #{m_pw} where m_id = #{m_id}
</update>
<select id="profilePhotoSelect" resultType="int" parameterType="String">
          select count(*) from profilephoto where pf_id = #{m_id}
</select>
<update id="fileUpdate" parameterType="HashMap">
          update profilephoto set pf_image = #{pf_image} where pf_id = #{pf_id}
</update>
<insert id="fileInsert" parameterType="HashMap">
          insert into profilephoto values(#{pf_id},#{pf_image})
</insert>
<update id="infoModifyCompanyUpdate" parameterType="company">
          Update company set c_bphone = #{c_bphone} where c_id = #{m_id}
</update>
<update id="infoModifyMemberUpdate" parameterType="company">
  update member set m_phone = #{m_phone},m_addr = #{m_addr},m_exaddr = #{m_exaddr} where m_id = #{m_id}
</update>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 패스워드 변경, 회원 정보 변경, .

## 33 JOB_NAME : 일반회원 내 프로그램페이지 이동 (일반회원관리 기본페이지) 작성자 : 윤상기

### ■ 개요

1. 자신이 결제한 내 프로그램, 일반피트니스 목록을 본다.

### ■ CONTROLLER

```
@RequestMapping(value = "/infoprogramn")
        public ModelAndView getInfoProgramN(String m_id) {
                mav=ys.programListN(m_id);
                return mav;
        }
```

1. 회원아이디(m_id)를 파라미터로 받음
2. yoonService 의 progrmListN(m_id)실행

### ■ SERVICE

```
public ModelAndView programListN(String m_id) {
                String view = null;
                List<Map<String, String>> getProgramListN = null;
                List<Map<String, String>> getNormalListN = null;
                getProgramListN = yDao.getproListN(m_id);
                getNormalListN = yDao.getnormalListN(m_id);
                String html = makeHTMLproPage(getProgramListN);
                String html2 = makeHTMLnorPage(getNormalListN);

                mav.addObject("programListN", html);
                mav.addObject("getnormalListN", html2);

                view = "manage/normal/normalMain";
                mav.setViewName(view);
                return mav;
        }
```

1. ArrayList 안에 HashMap 으로 getproListN,getnormalListN 에 실행된 결과값을 담는다
2. makeHTML 으로 결과값을 찍는다.
3. normalMain.jsp 으로 이동

### ■ DAO & MAPPER

```
List<Map<String, String>> getproListN(String m_id);
List<Map<String, String>> getnormalListN(String m_id);
```

**(1) DAO**

mapper 에 있는 getproListN, getnormalListN 로 파라미터 m_id 를 보냄

---

```
<select id="getproListN" parameterType="String" resultType="HashMap">
SELECT PS_MID,OP_TRAINER,OP_TIMES,OP_CLOCK,OP_DAY,DA_OPPERIOD ,OP_CODE,OP_CONTENT,PS_CODE ,AD_CODE,
AD_TITLE, AD_CATEGORY, OP_TRAINER, OP_PERIOD, AD_CATEGORY, PS_MID, DA_STATUS,OP_PERIOD
        FROM PS
        JOIN AD on PS_ADCODE = AD_CODE
        JOIN OP on PS_OPCODE = OP_CODE
```

```
            JOIN DA ON PS_CODE=DA_PSCODE
WHERE PS_MID= #{m_id} AND AD_CATEGORY NOT IN '일반'
</select>
<select id="getnormalListN" parameterType="String" resultType="HashMap">
            <![CDATA[
            SELECT PS_MID,M_ID,DA_OPPERIOD,OP_CODE,PS_CODE,AD_CODE, AD_TITLE, AD_STATUS, OP_PERIOD,
AD_CATEGORY, PS_MID, C_BNAME, DA_STATUS,OP_PERIOD
            FROM PS
            JOIN AD on PS_ADCODE = AD_CODE
            JOIN OP on PS_OPCODE = OP_CODE
            JOIN C ON AD_NAME=C_ID
            JOIN DA ON PS_CODE=DA_PSCODE
             JOIN M ON M_ID=AD_NAME
            WHERE PS_MID=#{m_id} AND AD_CATEGORY='일반'
            ]]>
</select>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 내 프로그램 정보를 Select 한다.

## ■ 결과

내 프로그램

| 광고명 | 트레이너 | 옵션 | 이용기간 | 카테고리 | 이용상태 | 출결현황 | 후기 |
|---|---|---|---|---|---|---|---|
| 동네 최대 휘트니스 | iciat5@never.com | 하는요일:화,시간:08:30~16:00,횟수:15 | 20190701~20190801 | PT | 이용중 | 출석체크확인 | 후기쓰기 |
| 필라테스 | iciat11@never.com | 하는요일:수,시간:12:00~13:00,횟수:12 | 20190701~20190826 | 필라테스 | 대기중 | 출석체크확인 | 후기쓰기 |
| 홈트레이닝 전문 | iciat11@never.com | 하는요일:월,시간:08:30~22:00,횟수:12 | 20190627~20190727 | 홈트레이닝 | 대기중 | 출석체크확인 | 후기쓰기 |

일반 피트니스

| 광고명 | 업체명 | 이용기간 | 출결확인 | 이용상태 | 후기쓰기 |
|---|---|---|---|---|---|
| 파라다이스 GYM | 파라다이스 | 20190627~20190727 | 이용중 | 출석체크확인 | 후기쓰기 |
| 부산최대 휘트니스 할인 이벤트 진행중 | 스타휘트니스 | 20190627~20190727 | 이용중 | 출석체크확인 | 후기쓰기 |

**상단에는 자신이 결제한 프로그램목록**

**하단에는 자신이 결제한 일반피트니스목록**

## 34 JOB_NAME : 내 출석 보기 작성자 : 윤상기

■ 개요

1. 해당 프로그램, 일반 피트니스에 자신의 출석체크목록을 확인한다.

■ CONTROLLER

```
@RequestMapping(value = "/calenderN")
        public ModelAndView calenderN(String ps_code) {
                mav=ys.calender(ps_code);
                return mav;
        }
```

1. 결제코드(ps_code)를 파라미터를 받음
2. yoonService 의 calender(ps_code)실행

■ SERVICE

```
public ModelAndView calender(String ps_code) {
                List<Map<String, String>> getDailyListN = null;
                Member sessionMb = (Member) session.getAttribute("mb");
                String m_id=sessionMb.getM_id();
                System.out.println();
                String view=null;
                Map<String,String> ck = new HashMap<>();
                ck.put("ps_code", ps_code);
                ck.put("m_id", m_id);
                getDailyListN=yDao.getDailyList(ck);
                String html = makeHTMLDailyPage(getDailyListN);
                mav.addObject("getDailyList", html);
                view="manage/calenderN";
                mav.setViewName(view);
                return mav; }
```

1. HashMap ck 에 ps_code, m_id 값을 담고 getDailyList(ck) 실행한값을 ArrayList 안에 HashMap 으로 결과값을 담는다.
2. makeHTML 으로 결과값을 찍는다.

■ DAO & MAPPER

```
List<Map<String, String>> getDailyList(Map<String, String> ck)
```

**(1) DAO**

mapper 에 있는 getDailyList 에 HashMap ck 를 보냄

---

```
<select id="getDailyList" parameterType="HashMap" resultType="HashMap">
SELECT DN_DATE FROM DN
JOIN DA on DN_DACODE = DA_CODE
JOIN PS on DA_PSCODE = PS_CODE
WHERE PS_MID= #{m_id} AND PS_CODE=#{ps_code}
</select>
```

**(2) MAPPER**

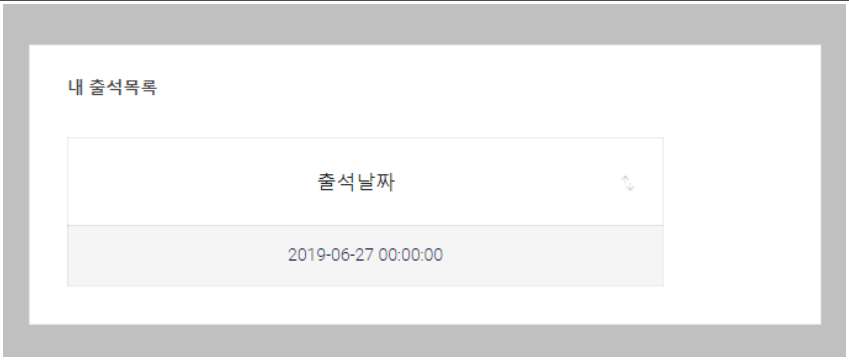
넘어온 파라미터 값을 이용해 해당 프로그램에 자신의 출석체크목록을 Select 한다.

■ FRONT-END

```
<div id="articleView_layer" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
        <div id="bg_layer"></div>
        <div id="contents_layer" style="width: 60em"></div>
<script>
function yyyyyy(bnum) {
        var list = new Array();
        var bbb=bnum;
        console.log(bbb);
        console.log(bbb);
        $('#articleView_layer').addClass('open');
        $.ajax({
                type:'get',
                url:'calenderN',
                data:{ps_code:bbb},
                dataType:'html',
                success:function(data){
                        $('#contents_layer').html(data);
                },
                error:function(error){
                        console.log(error);
                }
        });
}
</script>
```

해당 프로그램 출석체크확인 버튼을 누르면 ajax 으로 해당 url 이 컨트롤러에 실행되고 modelbox 로 출석체크목록이 나옴

■ 결과




**자신의 출석체크목록이 modelbox 로 나옴**

## 35 JOB_NAME : 일반회원 후기/문의 목록 작성자 : 윤상기

### ■ 개요

1. 자신이 작성한 후기 글과 문의 올린 글,답변을 본다.

### ■ CONTROLLER

```
@RequestMapping(value = "/questionlistn")
public ModelAndView questionlistN(String m_id) {
        mav=ys.questionlistN(m_id);
        return mav;
}
```

1. 회원아이디(m_id)를 파라미터를 받음
2. yoonService 의 questionlistN(m_id)실행

### ■ SERVICE

```
public ModelAndView questionlistN(String m_id) {
        String view = null;
        List<Map<String, String>> getReviewListN = null;
        List<Map<String, String>> getQuestionListN = null;
        getReviewListN = yDao.getRvListN(m_id);
        getQuestionListN = yDao.getQuestListN(m_id);
        String html =makeHTMLRVPage(getReviewListN);
        String html2 = makeHTMLQTPage(getQuestionListN);
        mav.addObject("reviewListN", html);
        mav.addObject("questionListN", html2);
        view = "manage/infoRVQAN";
        mav.setViewName(view);
        return mav;
}
```

1. ArrayList 안에 HashMap 으로 getRvListN(m_id),getQuestListN(m_id)에 실행된 결과값을 담는다
2. makeHTML 으로 결과값을 찍는다.
3. infoRVQAN.jsp 으로 이동

### ■ DAO & MAPPER

```
List<Map<String, String>> getRvListN(String m_id);
List<Map<String, String>> getQuestListN(String m_id);
```

**(1) DAO**

mapper 에 있는 getRvListN,getQuestListN 에 각각 파라미터 m_id 를 보냄

---

```
<select id="getRvListN" parameterType="String" resultType="HashMap">
        <![CDATA[
    SELECT RV_NUM,PS_CODE,PS_CODE,RV_CONTENT,RV_DATE,RV_STPOINT,PS_ADCODE,AD_CODE,OP_CONTENT,AD_TITLE
    FROM RV
    JOIN PS ON RV_PSCODE=PS_CODE
    JOIN OP ON PS_OPCODE=OP_CODE
    JOIN AD ON OP_ADCODE=AD_CODE
    WHERE RV_NAME=#{m_id}
```

```
    ORDER BY RV_DATE DESC
    ]]>
</select>
<select id="getQuestListN" parameterType="String" resultType="HashMap">
<![CDATA[
SELECT
QA_NUM,QA_ADCODE,QA_WRITER,QA_ANSWER,QA_WCONTENT,QA_WDATE,QA_ACONTENT,QA_ADATE,QA_TRUE,AD_TITLE
FROM QA
JOIN AD ON QA_ADCODE=AD_CODE
WHERE QA_WRITER =#{m_id}
ORDER BY QA_ADATE DESC
 ]]>
</select>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 자신이 쓴 후기 글과 문의 글과 답변을 Select 한다.

## ■ FRONT-END

```
<div class="modal-body">
문의 내용 :
<input id="qa_num" type="hidden"/>
<div id="qa_wcontent" style="text-align: center;margin: 50px;"></div>
</div>
<div class="modal-body">
<input type="text" id="qa_acontent" name="qa_acontent" placeholder="아직 답변이 없습니다." disabled="disabled"
style="text-align: center; width: 750px; height: 150px;" />
</div>
```

문의 글 내용보기버튼을 누르면 ajax 으로 해당 url 이 컨트롤러에 실행되고 modelbox 로 답변글이 나옴
문의 글에 아직 답변을 안했으면 -아직 답변이 없습니다.-라고 나옴

## ■ 결과

| 광고명 | 옵션명 | 내용 | 올린시간 |
|---|---|---|---|
| 필라테스 | 초급 | 후기입니다.11111 | 2019-07-08 11:41:20 |
| 동네 최대 휘트니스 | 제이팍트레이너 | 선생님이 좀 못생겼어요 | 2019-06-27 15:47:02 |

| 광고명 | 내용보기 | 올린시간 | 답변시간 |
|---|---|---|---|
| 필라테스 | 내용보기 | 2019-06-27 19:38:18 | 2019-06-27 20:02:56 |
| 동네 최대 휘트니스 | 내용보기 | 2019-06-27 17:49:54 | 2019-06-27 17:49:54 |
| 동네 최대 휘트니스 | 내용보기 | 2019-06-27 17:42:38 | 2019-06-27 17:42:38 |
| 부산최대 휘트니스 할인 이벤트 진행중 | 내용보기 | 2019-06-27 16:54:08 | 2019-06-27 16:54:08 |

**상단에는 자신이 쓴 후기 글 목록, 하단에는 자신이 쓴 문의 글 목록**

문의 내용 :

카드 되나요???

10% 금액 추가됩니다

**자신이 쓴 문의 내용과 문의 답변**

| 36 | JOB_NAME : 후기 글쓰기 페이지 | 작성자 : 윤상기 |
|---|---|---|

■ 개요

1. 자신이 결제한 프로그램에 후기 글쓰기 페이지로 이동

■ CONTROLLER

```
@RequestMapping(value = "/reviewwritefrm")
        public ModelAndView reviewWriteFrm(String m_id, String ps_code)  {
                mav=ys.reviewWriteFrm(m_id, ps_code);
                return mav;
        }
```

1. 회원아이디(m_id)와 결제코드(ps_code)를 파라미터를 받음
2. yoonService 의 reviewWriteFrm(m_id, ps_code)실행

■ SERVICE

```
public ModelAndView reviewWriteFrm(String m_id, String ps_code) {
                ModelAndView mav = new ModelAndView();
                Review rv = new Review();
                rv.setRv_name(m_id);
                rv.setRv_pscode(ps_code);
                if (yDao.reviewOverlap(rv)) {// true
                        String alert = "alert('이미 해당 광고글에 후기등록을 하셨습니다');";
                        mav.addObject("alert", alert);
                        mav.setViewName("forward:/infoprogramn");
                } else {// false
                        mav.addObject("m_id", m_id);
                        mav.addObject("ps_code", ps_code);
                        mav.setViewName("manage/review");
                }
                return mav;
        }
```

1. ReviewBean 에 넘겨 받은 데이터 값을 담는다.
2. reviewOverlap 실행된 결과값으로 자신이 해당 광고 글에 후기 등록을 했는지 유효성 검사
3. 해당 광고 글에 등록된 자신이 쓴 후기 글이 없으면 m_id, ps_code 데이터를 가져가고 Review.jsp 로 이동

■ DAO & MAPPER

```
boolean reviewOverlap(Review rv);
```

**(1) DAO**

mapper 에 있는 reviewOverlap 에 Review 를 보냄

---

```
<select id="reviewOverlap" parameterType="review" resultType="boolean">
        SELECT COUNT(*) FROM REVIEW WHERE RV_NAME=#{rv_name} AND RV_PSCODE=#{rv_pscode}
```

**(2) MAPPER**

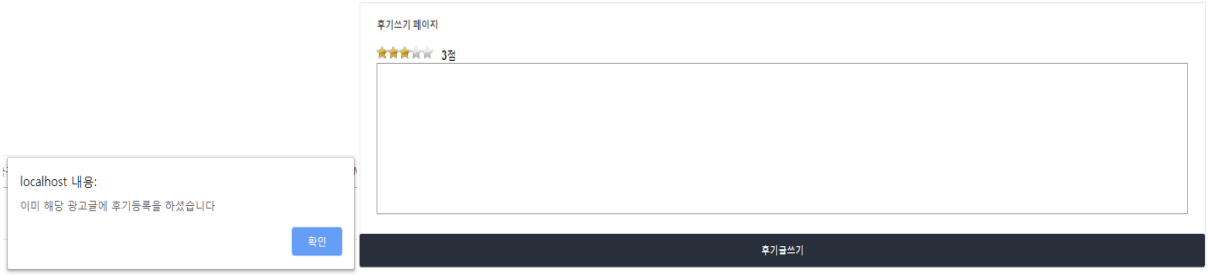
넘어온 파라미터 값을 이용해 해당 광고 글에 자신이 쓴 후기 글을 Select 한다.

## ■ FRONT-END

```
<script>
//star rating
var starRating = function(){
  var $star = $(".star-input"),
      $result = $star.find("output>b");
  $(document)
    .on("focusin", ".star-input>.input", function(){
    $(this).addClass("focus");
  })
    .on("focusout", ".star-input>.input", function(){
    var $this = $(this);
    setTimeout(function(){
      if($this.find(":focus").length === 0){
        $this.removeClass("focus");
      }
    }, 100);
  })
    .on("change", ".star-input :radio", function(){
    $result.text($(this).next().text());
  })
    .on("mouseover", ".star-input label", function(){
    $result.text($(this).text());
  })
    .on("mouseleave", ".star-input>.input", function(){
    var $checked = $star.find(":checked");
    if($checked.length === 0){
      $result.text("0");
    } else {
      $result.text($checked.next().text());
    }
  });
};
starRating();
</script>
```

Ridio 로 1~5 점 후기점수 제어 하고 별로 표기하는 function

## ■ 결과




**해당 광고글에 이미 한번 후기등록했으면 해당 alert 창이 나온다.**

**후기등록한게 없으면 후기쓰기 페이지로 이동.**

## 37 JOB_NAME : 후기 글 등록 작성자 : 윤상기

### ■ 개요

1. 후기 글 등록하기

### ■ CONTROLLER

```
@RequestMapping(value = "/reviewwriteinsert")
        public ModelAndView insertReview(Review rv) {
                mav=ys.insertReview(rv);
                return mav;
        }
```

1. Review bean 에 저장된 ps_code(결제코드), m_id(회원아이디)있어서 파라미터로 Review rv 를 가져감
2. yoonService 의 insertReview(rv)실행

### ■ SERVICE

```
public ModelAndView insertReview(Review rv) {
                String view = null;
                Member sessionMb = (Member) session.getAttribute("mb");
                String m_id=sessionMb.getM_id();
                boolean insertRv = yDao.reviewInsert(rv);
                view = "forward:questionlistn?m_id="+m_id+"";
                mav.setViewName(view);
                return mav;
        }
```

1. 가져온 파라미터와 후기글쓴거, 별점을 가지고 reviewInsert(rv) 실행시킴
2. 후기글쓰기를 완료하면 url:questionlistn(일반회원 후기/문의관리)페이지로 파라미터 m_id 가지고 포워딩

### ■ DAO & MAPPER

```
boolean reviewInsert(Review rv);
```

**(1) DAO**

mapper 에 있는 reviewInsert 에 Review 를 보냄

---

```
<insert id="reviewInsert" parameterType="review">
        INSERT INTO REVIEW VALUES(RV_SEQ.NEXTVAL,#{rv_pscode},#{rv_name},#{rv_content},sysdate,#{rv_stpoint})
</insert>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 해당 광고 글에 자신이 쓴 후기 글을 Insert 한다.

### ■ FRONT-END

```
<form action="reviewwriteinsert" method="post">
```

쓴 후기글과 별점을 form action 으로 컨트롤러에 넘긴다.

### ■ 결과

**해당 광고에 정상적으로 후기 등록했으면 해당 alert 창이 나오고 해당 광고에 후기 글이 등록됨.**

## 38 JOB_NAME : 내 찜 목록 내역 작성자 : 윤상기

### ■ 개요

1. 자신이 찜 한 광고 글 목록을 본다.

### ■ CONTROLLER

```
@RequestMapping(value = "/dibsn")
public ModelAndView dibsList(String m_id) {
        System.out.println("mmmmmmmmmmmmmmmm="+m_id);
        mav=ys.dibsList(m_id);
        return mav; }
```

1. 회원아이디(m_id)를 파라미터를 받음
2. yoonService 의 dibsList(m_id)실행

### ■ SERVICE

```
public ModelAndView dibsList(String m_id) {
        String view = null;
        List<Map<String, String>> dibs = null;
        dibs = yDao.getdibs(m_id);
        String html = makeHTMLdibsPage(dibs);
        mav.addObject("dibs", html);
        view = "manage/dibsListN";
        mav.setViewName(view);
        return mav; }
```

1. ArrayList 안에 HashMap 으로 getdibs(m_id)에 실행된 결과값을 담는다
2. makeHTML 으로 결과값을 찍는다.
3. dibsListN.jsp 으로 이동

### ■ DAO & MAPPER

```
List<Map<String, String>> getdibs(String m_id);
```

**(1) DAO**

mapper 에 있는 getdibs 에 파라미터 m_id 를 보냄

---

```
<select id="getdibs" resultType="HashMap">
SELECT AD_TITLE,AD_CODE FROM AD JOIN DIBS ON AD_CODE=D_ADCODE WHERE D_ID=#{m_id}
</select>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 자신이 찜 한 광고 글 목록을 Select 한다.

### ■ 결과

| 광고명 | 삭제 |
|---|---|
| 부산최대 휘트니스 할인 이벤트 진행중 | 삭제 |
| 여기가 바로 그헬스장 | 삭제 |
| 우리동네 요가왕 오전반모집 | 삭제 |
| 우리모두 GYM가서 예뻐GYM | 삭제 |
| 자 떠나자~ 헬스장으로~ | 삭제 |
| 파라다이스 GYM | 삭제 |

## 39 JOB_NAME : 내 결제 내역 보기 작성자 : 윤상기

### ■ 개요

1. 자신이 결제한 결제내역목록을 본다.

### ■ CONTROLLER

```
@RequestMapping(value = "/payhistoryn")
public ModelAndView payList(String m_id) {
        mav=ys.payListN(m_id);
        return mav;
}
```

1. 회원아이디(m_id)를 파라미터를 받음
2. yoonService 의 payListN(m_id)실행

### ■ SERVICE

```
public ModelAndView payListN(String m_id) {
        String view = null;
        List<Map<String, String>> getPayListN = null;
        getPayListN = yDao.getpayListN(m_id);
        String html = makeHTMLpayPage(getPayListN);
        mav.addObject("payListN", html);
        view = "manage/payHistoryN";
        mav.setViewName(view);
        return mav;
}
```

1. ArrayList 안에 HashMap 으로 getpayListN(m_id)에 실행된 결과값을 담는다
2. makeHTML 으로 결과값을 찍는다.
3. payHistoryN.jsp 으로 이동

### ■ DAO & MAPPER

```
List<Map<String, String>> getpayListN(String m_id);
```

**(1) DAO**

mapper 에 있는 getpayListN 에 파라미터 m_id 를 보냄

---

```
<select id="getpayListN" parameterType="String" resultType="HashMap">
SELECT DA_OPPERIOD,AD_TITLE,AD_CATEGORY, PS_PRICE, PS_DATE, OP_CONTENT, OP_PERIOD FROM PS
JOIN AD on PS_ADCODE = AD_CODE
JOIN OP on OP_CODE = PS_OPCODE
JOIN DA ON DA_PSCODE =PS_CODE WHERE PS_MID=#{m_id}
</select>
```

**(2) MAPPER**

넘어온 파라미터 값을 이용해 자신이 결제한 목록을 Select 한다.

### ■ 결과

| 광고명 | 카테고리 | 옵션 | 이용기간 | 결제금액 | 결제날짜 |
|---|---|---|---|---|---|
| 동네 최대 휘트니스 | PT | 제이딱트레이너 | 20190701~20190801 | 300000 | 2019-06-27 15:41:57 |
| 부산최대 휘트니스 할인 이벤트 진행중 | 일반 | 일반+케비넷 | 20190627~20190727 | 70000 | 2019-06-27 17:45:45 |
| 파라다이스 GYM | 일반 | 일반 | 20190627~20190727 | 80000 | 2019-06-27 17:34:35 |

## 40 JOB_NAME : 내 정보보기 페이지 이동 작성자 : 윤상기

■ 개요

1. 자신의 회원정보를 볼 수 있다.

■ CONTROLLER

```
@RequestMapping(value = "/infomodifyfrmn")
public ModelAndView modifyN(String m_id) {
        mav=ys.modifyN(m_id);
        return mav;
}
```

1. 회원아이디(m_id)를 파라미터를 받음
2. yoonService 의 modifyN(m_id)실행

■ SERVICE

```
public ModelAndView modifyN(String m_id){
        String view = null;
        Member mb = yDao.getModifyN(m_id);
        Member mbPhoto = yDao.getPhotoModifyN(m_id);
        mav.addObject("mb", mb);
        mav.addObject("mbPhoto", mbPhoto);
        view = "manage/infoModifyN";
        mav.setViewName(view);
        return mav;
}
```

1. Member bean 에 getModifyN(m_id), getPhotoModifyN(m_id)에 실행된 결과값을 담는다
2. jsp 에서 각각 원하는 위치에 EL 로 결과값을 표현한다.
3. infoModifyN.jsp 으로 이동

■ DAO & MAPPER

```
Member getModifyN(String m_id);
Member getPhotoModifyN(String m_id);
```

**(1) DAO**

mapper 에 있는 getModifyN,getphotoModifyN 에 각각 파라미터 m_id 를 보냄

---

```
<select id="getModifyN" parameterType="String" resultType="Member">
SELECT M_ID, M_PW ,M_NAME, M_PHONE, TO_CHAR(M_BIRTH,'YYYY/MM/DD') as M_BIRTH,
M_ADDR, M_EXADDR, M_KIND FROM
M WHERE M_ID = #{m_id}
</select>
<select id="getPhotoModifyN" parameterType="String" resultType="Member">
SELECT * FROM PF WHERE PF_ID=#{m_id}
</select>
```

**(2) MAPPER**

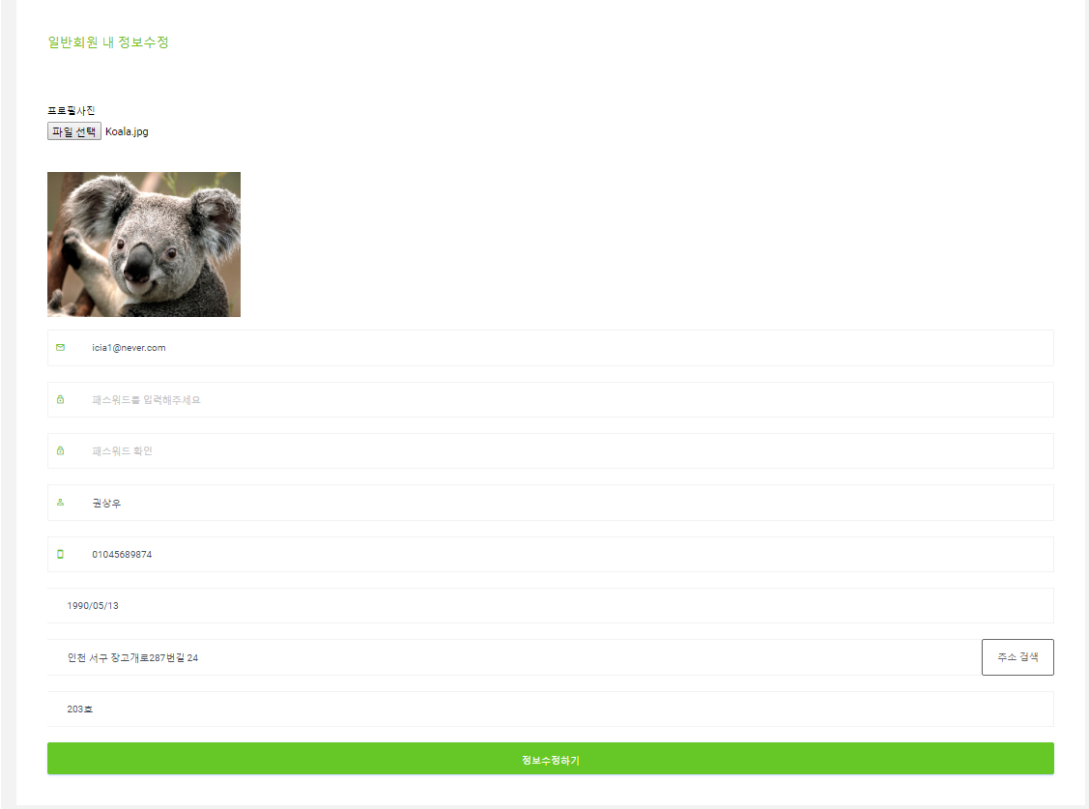
넘어온 파라미터 값을 이용해 자신의 회원정보와 프로필 사진을 Select 한다.

## ■ FRONT-END

```
<input type="file" id="input_img" name="pf_image" />

var sel_file;
	$(document).ready(function() {
		$("#input_img").on("change", handleImgFileSelect);
	});
	function handleImgFileSelect(e) {
		var files = e.target.files;
		var filesArr = Array.prototype.slice.call(files);
		filesArr.forEach(function(f) {
			if (!f.type.match("image.*")) {
				alert("확장자는 이미지 확장자만 가능합니다.");
				return;
			}
			sel_file = f;
			var reader = new FileReader();
			reader.onload = function(e) {
				$("#img").attr("src", e.target.result);
			}
			reader.readAsDataURL(f);
		});
	}
```

Input type=file 으로 자신의 원하는 프로필사진을 선택하면 id="input_img"영역에 미리 보기로 사진을 볼 수 있다.

## ■ 결과



## 41 JOB_NAME : 내 정보 수정 작성자 : 윤상기

■ 개요

1. 자신의 회원정보를 수정 할 수 있다.

■ CONTROLLER

```
@RequestMapping(value = "/infomodifyn")
public ModelAndView infomodifyn(MultipartHttpServletRequest multi) {
        mav=ys.infomodifyn(multi);
        return mav;
}
```

1. 파일 업로드(프로필사진)을 하기 위해 MultipartHttpServletRequest 클래스를 이용
2. yoonService 의 infomodifyn(multi)실행

■ SERVICE

```
public ModelAndView infomodifyn(MultipartHttpServletRequest multi) {
        String view=null;
        Member mb=new Member();

        Member sessionMb = (Member) session.getAttribute("mb");
        String m_id=sessionMb.getM_id();
        String m_birth=sessionMb.getM_birth();
        String m_name=sessionMb.getM_name();
        String m_pw=multi.getParameter("m_pw");
        String m_phone=multi.getParameter("m_phone");
        String m_addr=multi.getParameter("m_addr");
        String m_exaddr=multi.getParameter("m_exaddr");
        String pf_image=multi.getParameter("pf_image");

        int i=yDao.imgOverlap(m_id);
        if(i==0) {
                upload.fileUp(multi, m_id);
                BCryptPasswordEncoder pwdEncoder = new BCryptPasswordEncoder();
                mb.setM_id(m_id);
                mb.setM_pw(pwdEncoder.encode(m_pw));
                mb.setM_phone(m_phone);
                mb.setM_addr(m_addr);
                mb.setM_exaddr(m_exaddr);
                mb.setM_birth(m_birth);
                mb.setM_name(m_name);
                mb.setPf_image(pf_image);
                yDao.updateNorMb(mb);
                Member mb1=yDao.getModifyN(m_id);
                Member mbPhoto = yDao.getPhotoModifyN(m_id);
                mav.addObject("mb", mb1);
                mav.addObject("mbPhoto", mbPhoto);
                return mav;
```

```
}else if(i>=1) {
        upload.fileUp2(multi, m_id);
        BCryptPasswordEncoder pwdEncoder = new BCryptPasswordEncoder();
        mb.setM_id(m_id);
        mb.setM_pw(pwdEncoder.encode(m_pw));
        mb.setM_phone(m_phone);
        mb.setM_addr(m_addr);
        mb.setM_exaddr(m_exaddr);
        mb.setM_birth(m_birth);
        mb.setM_name(m_name);
        mb.setPf_image(pf_image);
        yDao.updateNorMb(mb);
        Member mb1=yDao.getModifyN(m_id);
        Member mbPhoto = yDao.getPhotoModifyN(m_id);
        mav.addObject("mb", mb1);
        mav.addObject("mbPhoto", mbPhoto);
        return mav;
}

    return mav;
}

public class UploadFile {
    //파일 업로드 메소드
    //String fullPath="D:/Work/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps/SpringMVC-Board/resources/upload";
    @Autowired
    private YoonDao yDao;

    public boolean fileUp(MultipartHttpServletRequest multi, String m_id){
        System.out.println("fileUp");
        //1.이클립스의 물리적 저장경로 찾기 (이클립스가 가상으로 경로지정해줌)
        String root=multi.getSession().getServletContext().getRealPath("/");
        System.out.println("root="+root);
        String path=root+"resources/upload/";
        //2.폴더 생성을 꼭 할것...
        File dir=new File(path);
          if(!dir.isDirectory()){ //upload폴더 없다면 dir.mkdir(); //upload폴더 생성
          dir.mkdirs(); //s 안붙이면 딱 그폴더만 생성되고 s붙이면 경로 상 상위폴더까지 같이만들어줌
          }
        //3.파일을 가져오기-파일태그 이름들 반환
        Iterator<String> files=multi.getFileNames(); //파일업로드 2개이상일때
        Map<String,String> fMap=new HashMap<String, String>();
        fMap.put("m_id", m_id); //글번호 저장
        boolean f=false;
        while(files.hasNext()){
            String fileTagName=files.next();
            System.out.println("fileTag="+fileTagName);
```

```
//파일 메모리에 저장
MultipartFile mf=multi.getFile(fileTagName); //실제파일
String oriFileName=mf.getOriginalFilename();   //a.txt 오리지날파일명
fMap.put("pf_image", oriFileName); //DB에 저장
//4.시스템파일이름 생성   a.txt   ==>112323242424.txt
String sysFileName=System.currentTimeMillis()+"." //현재시간
+oriFileName.substring(oriFileName.lastIndexOf(".")+1);
//.확장자를 찾아서 확장자앞에다가currentTimeMillis붙임
fMap.put("pf_image", sysFileName);
//5.메모리->실제 파일 업로드
try {
        mf.transferTo(new File(path+sysFileName)); //실제 파일 업로드부분
        f=yDao.fileInsert(fMap); //업로드한 파일 이름들 DB에 넣기
        f=yDao.fileUpdate(fMap);
}catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
} //while End
if(f)
        return true;
return false;
}

public boolean fileUp(MultipartHttpServletRequest multi, String m_id){
        System.out.println("fileUp");
        //1.이클립스의 물리적 저장경로 찾기 (이클립스가 가상으로 경로지정해줌)
        String root=multi.getSession().getServletContext().getRealPath("/");
        System.out.println("root="+root);
        String path=root+"resources/upload/";
        //2.폴더 생성을 꼭 할것...
        File dir=new File(path);
          if(!dir.isDirectory()){ //upload폴더 없다면 dir.mkdir(); //upload폴더 생성
          dir.mkdirs(); //s 안붙이면 딱 그폴더만 생성되고 s붙이면 경로 상 상위폴더까지 같이만들어줌
          }
        //3.파일을 가져오기-파일태그 이름들 반환
        Iterator<String> files=multi.getFileNames(); //파일업로드 2개이상일때
        Map<String,String> fMap=new HashMap<String, String>();
        fMap.put("m_id", m_id); //글번호 저장
        boolean f=false;
        while(files.hasNext()){
                String fileTagName=files.next();
                System.out.println("fileTag="+fileTagName);
                //파일 메모리에 저장
                MultipartFile mf=multi.getFile(fileTagName); //실제파일
                String oriFileName=mf.getOriginalFilename();   //a.txt 오리지날파일명
                fMap.put("pf_image", oriFileName); //DB에 저장
                //4.시스템파일이름 생성   a.txt   ==>112323242424.txt
```

```
                    String sysFileName=System.currentTimeMillis()+"." //현재시간
                    +oriFileName.substring(oriFileName.lastIndexOf(".")+1);
                    //.확장자를 찾아서 확장자앞에다가currentTimeMillis붙임
                    fMap.put("pf_image", sysFileName);
                    //5.메모리->실제 파일 업로드
                    try {
                        mf.transferTo(new File(path+sysFileName)); //실제 파일 업로드부분
                        f=yDao.fileInsert(fMap); //업로드한 파일 이름들 DB에 넣기
                        f=yDao.fileUpdate(fMap);
                    }catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                } //while End
                if(f)
                    return true;
                return false;
        }
```

1. 변경된 정보들을 변수에 담고 아이디,이름,생일은 변경 할 수 없어서 세션에 저장 된 값만 가져옴
2. 등록되어 있는 프로필 사진이 있는지 유효성 검사를 위해 imgOverlap(m_id)을 실행
3. 등록되어 있는 프로필 사진이 있으면 fileUpdate(fMap)실행, 프로필 사진이 없으면 fileInsert(fMap)실행

## ■ DAO & MAPPER

```
boolean fileInsert(Map<String, String> fMap);
boolean fileUpdate(Map<String, String> fMap);
```

**(1) DAO**

mapper 에 있는 getDailyList 에 HashMap fMap 를 보냄

---

```
<select id="imgOverlap" parameterType="String" resultType="Integer">
        SELECT COUNT(*) FROM PF WHERE PF_ID=#{m_id}
</select>
<insert id="fileInsert" parameterType="String">
        INSERT INTO PROFILEPHOTO VALUES(#{m_id},#{pf_image})
</insert>
<update id="fileUpdate" parameterType="String">
UPDATE PROFILEPHOTO SET PF_IMAGE = #{pf_image} WHERE PF_ID=#{m_id}
</update>
<update id="updateNorMb" parameterType="Member">
UPDATE MEMBER SET M_PW = #{m_pw}, M_PHONE = #{m_phone}, M_ADDR = #{m_addr}, M_EXADDR= #{m_exaddr}
WHERE M_ID=#{m_id}
</update>
```

**(2) MAPPER**

1. id="imgOverlap"에선 넘어온 파라미터 값으로 프로필사진이 등록 되어 있는지 COUNT
2. id="fileInsert"에선 프로필사진이 없을 경우 실행, 있을 경우 id="fileUpdate"을 실행
3. id="updateNorMb"에선 자신이 바꾼 정보를 UPDATE 실행

## 42 JOB_NAME : 회원 프로필 보기 작성자 : 장기림

■ 개요

1. 어느 페이지에서든 Request 된 회원아이디를 통해 프로필 종류를 나누고 프로필 정보 Response 한다.

■ CONTROLLER

```
@RequestMapping(value = "/profilepage",produces ="application/json; charset=utf-8", method
= RequestMethod.GET)
        @ResponseBody
        public String profilePage(String m_id) {
                System.out.println("url:/profilepage");
                html = ks.profilePage(m_id);
                return html;
        }
```

■ SERVICE

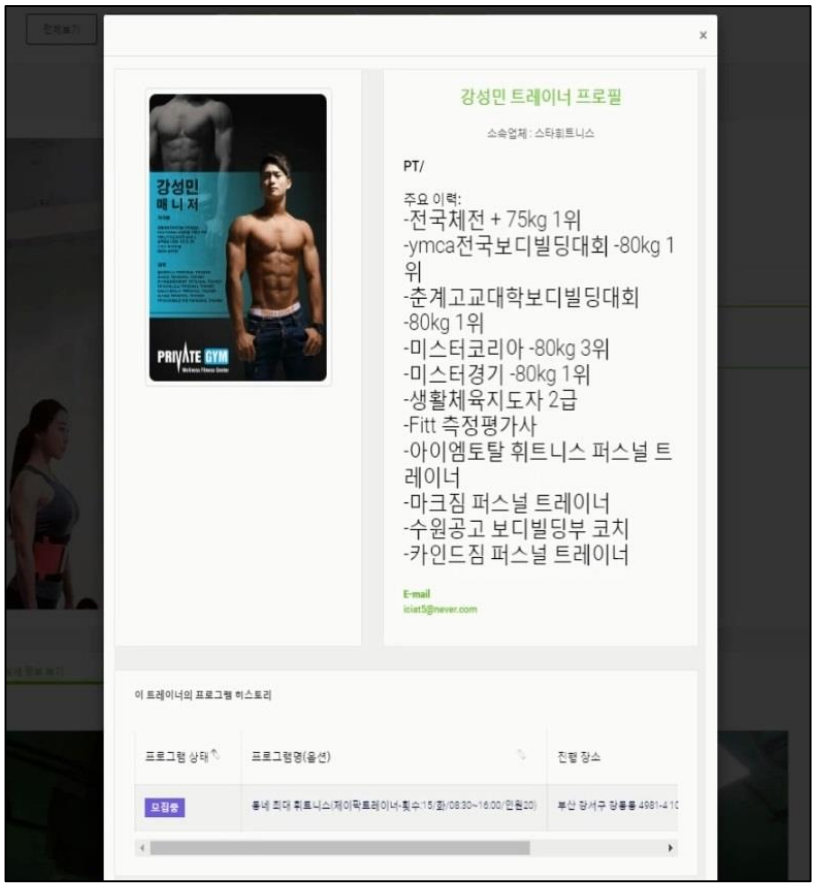
```
public String profilePage(String m_id) {
                Member mb=null;
                String html = null;
                Map<String,String> tp=null;
                List<Map<String, String>> tpo=null;
                List<Map<String, String>> tpc=null;
                Map<String, String> cp=null;
                List<Map<String, String>> cpo=null;
                String kind = kDao.getMemberKind(m_id);
                if(kind.equals("n")) {
                        mb=kDao.getNormalInfo(m_id);
                        html = makeHTMLNormalProfile(mb);
                }else if(kind.equals("t")) {
                        tp=kDao.getTrainerProfile(m_id);
                        tpo=kDao.getTrainerPOption(m_id);
                        tpc=kDao.getTrainerPCategory(m_id);
                        html = makeHTMLTrainerProfile(tp, tpo, tpc);
                }else if(kind.equals("c")) {
                        cp=kDao.getCompanyProfile(m_id);
                        cpo=kDao.getCompanyProfileOption(m_id);
                        html = makeHTMLCompanyProfile(cp,cpo);
                }
                return html;
        }
```

■ DAO & MAPPER

```
public Map<String, String> getTrainerProfile(String m_id);
<select parameterType="String" id="getTrainerProfile" resultType="HashMap">
                SELECT M_ID, M_NAME, M_KIND, T_CID, NVL(T_CAREER,'none') as T_CAREER,
C_BNAME, NVL(PF_IMAGE,'none') as PF_IMAGE
                FROM M
                LEFT JOIN T ON M_ID=T_ID
                LEFT JOIN C ON T_CID=C_ID
                LEFT JOIN PF ON M_ID=PF_ID
                WHERE M_ID=#{m_id}
</select>
public List<Map<String, String>> getTrainerPOption(String m_id);
<select parameterType="String" id="getTrainerPOption" resultType="HashMap">
                SELECT OP_CODE, OP_ADCODE, OP_TRAINER, OP_CONTENT, OP_PERIOD, OP_TIMES,
OP_DAY, OP_CLOCK, OP_PERSONNEL,
                AD_TITLE, AD_STATUS, AD_DATE, MC.M_ADDR AS AD_ADDR, MT.M_ADDR AS T_ADDR
                FROM OP
                LEFT JOIN AD ON OP_ADCODE = AD_CODE
```

```
                    LEFT JOIN M MT ON OP_TRAINER=MT.M_ID
                    LEFT JOIN M MC ON AD_NAME=MC.M_ID
                    LEFT JOIN C ON AD_NAME=C_ID
                    WHERE OP_TRAINER=#{m_id}
</select>
public List<Map<String, String>> getTrainerPCategory(String m_id);
<select parameterType="String" id="getTrainerPCategory" resultType="HashMap">
                    SELECT DISTINCT AD_CATEGORY
                    FROM OP
                    LEFT JOIN AD ON OP_ADCODE = AD_CODE
                    WHERE OP_TRAINER=#{m_id}
</select>
```

■ 결과




-트레이너 프로필 예시-