

# 웹 프로그래밍 포트폴리오

권형민

프로젝트 명 : 장물나라

프로젝트 개요 : 경매 사이트

주요 기능 : 회원 가입, 이메일 인증, 경매 글 작성, 입찰, 낙찰, 유찰, 관리자 기능(글 임의 삭제 가능), 자신이 등록한 경매 및 입찰한 경매 관리.

개발환경 : jdk1.8, eclipse luna, tomcat 8.0, oracle 10g, spring framework 4.1.7

본인이 맡은 기능 : 회원 가입, 이메일 인증 과정(메일 서버는 다른 팀원이 맡음.)

자신이 등록 또는 입찰한 경매 관리 기능,

관련 소스코드

## 로그인 관련

### 로그인 핸들러

패키지 및 임포트 생략

@Controller

```
public class LoginProHandler implements CommandHandler {
    @Resource(name="logondao")
    Dao dao;
    @RequestMapping("/loginPro")
    @Override
    public ModelAndView process(HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        request.setCharacterEncoding("utf-8");
        String email = request.getParameter("email");
        String passwd = request.getParameter("passwd");

        int result = dao.check(email, passwd);
        LogonDataBean dto = dao.getMember(email);
        if(result==1){ //1일 경우 이메일, 비밀번호가 일치하는 항목이 하나 있는 경우를
            뜻함.
                //세션에 계정정보 저장
                request.getSession().setAttribute("memId", email);
                request.getSession().setAttribute("gold", dto.getGold());
                request.getSession().setAttribute("name", dto.getName());
                request.getSession().setAttribute("rank", dto.getRank());
            }
        return new ModelAndView("logon/loginPro", "result", result);
    }
}
```

## 이메일 인증 관련

자바 스크립트

```
function confirmEmail(){
    var email=document.confirmEmailForm.email.value
    if(!document.confirmEmailForm.email.value)
    {
        . . . //예외처리들
    }
    else
    {
        var url=emailConfirmCheck.do?email="+document.confirmEmailForm.email.value;
        window.open( url, "randomGen", "menubar=no, toolbar=no, scrollbar=no, statusbar=no,
        width=300, height=200" );
    }
}
url에 해당하는 핸들러
```

@Controller

```
public class EmailConfirmCheckHandler implements CommandHandler {
    @Autowired
    private JavaMailSender mailSender;
```

```

@Resource(name="logondao")
private Dao dao;
private Random rand=new Random();
private String code=new String();
private Map<String, Object> map=new HashMap<String, Object>();
@RequestMapping("emailConfirmCheck")
@Override
public ModelAndView process(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    request.setCharacterEncoding("UTF-8");
    String email=request.getParameter("email");
    int result=dao.checkMember(email);
    map.put("result", result);
    if(result>0) //이메일이 이미 있을 경우 중복된 이메일 메시지.
    {

        return new ModelAndView("logon/emailConfirmCheck","result", result);
    }
    else if(result==0)
    {
        code="";
        code+=rand.nextInt(10);
        code+=rand.nextInt(10);
        code+=rand.nextInt(10);
        code+=rand.nextInt(10);
        //코드를 랜덤으로 4자리 생성
        map.put("code", code);
        //send email contain code;
        MimeMessage mailer = mailSender.createMimeMessage();
        try {
            MimeMessageHelper messageHelper = new MimeMessageHelper(mailer,
true, "UTF-8");

            messageHelper.setSubject("장물나라 인증번호");
            messageHelper.setTo(email);
            messageHelper.setFrom("khpmttest@gmail.com");
            messageHelper.setText("인증번호 : " + code);
            mailSender.send(mailer);
            //이메일 발송
        } catch (MessagingException e) {
            e.printStackTrace();
        }

        return new ModelAndView("logon/emailConfirmCheck",map);
        //코드 정보를 확인창에 보냄
    }
    return new ModelAndView("default");
}
}

```

자신의 경매 관리 페이지에 대한 핸들러  
 조금 길기 때문에 진행중인, 자신이 등록한 경매에 대한 코드만 기록  
 //자신이 진행하는 경매, 입찰한 경매에 대한 정보를 다루는 핸들러. 게시판 소스코드를  
 참조함.

```

@Controller
public class MyAuctionListHandler implements CommandHandler {
    @Resource(name="logondao")
    Dao dao;
    @Resource(name="auctiondao")
    auctionboard.Dao auctiondao;
    @RequestMapping("/myAuctionList")
    @Override
    public ModelAndView process(HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        request.setCharacterEncoding("utf-8");
    }
}

```

```

String email=(String)request.getSession().getAttribute("memId");
request.setAttribute("commission", dao.getCommission(email));
//System.out.println(email);
if(email==null || email.equals(""))
{
    return new ModelAndView("default");
}
else
{
    //현재 경매에 대한 변수
    int count = 0;
    String pageNum = null;
    int currentPage = 0;
    int start = 0;
    int end = 0;
    int number = 0;
    int pageCount = 0;
    int startPage = 0;
    int endPage = 0;

    int pageSize = 10;
    int pageBlock = 5;
    //=====for old auctions===== 지난 경매에 대한 변수
    . . .

    String mode=request.getParameter("mode");
    if(mode==null || mode.equals("") || mode.equals("sell"))
//요청 페이지가 내가 판매하는 경매건일 경우
    {
        count=dao.getMyAuctionCount(email);
        pageNum=request.getParameter("pageNum");
        if(pageNum==null || pageNum.equals(""))
        {
            pageNum="1";
        }
        try{
            currentPage=Integer.parseInt(pageNum);
        }catch(NumberFormatException e)
        {
            return new ModelAndView("default");
        }
        start=(currentPage-1)*pageSize+1;
        end=start+pageSize-1;
        number=count-(currentPage-1)*pageSize;
        if(count > 0)
        {
            pageCount=(count/pageSize)+(count%pageSize>0?
1:0);

            if(currentPage>pageCount)
            {
                currentPage=pageCount;
            }
            startPage=(currentPage/pageBlock)*pageBlock+1;
            if(currentPage%pageBlock==0)
            {
                startPage-=pageBlock;
            }
            endPage=startPage+pageBlock-1;

```

```

        if(endPage>pageCount) endPage=pageCount;
        Map<String, Object> map = new HashMap<String,
Object>();
        map.put("start", start);
        map.put("end", end);
        map.put("email", email);
        List<AuctionDataBean>
myAuctions=dao.getMyAuctions(map);
        request.setAttribute("myAuctions", myAuctions);
        //System.out.println("list:"+myAuctions.size());
    }
    //=====for old auctions===== 지난 경매
    . . . //마찬가지로 표현
    Map<String, Object> model=new HashMap<String,
Object>();
        model.put("count", count);
        model.put("pageNum", pageNum);
        model.put("currentPage", currentPage);
        model.put("number", number);
        model.put("pageCount", pageCount);
        model.put("startPage", startPage);
        model.put("endPage", endPage);
        model.put("pageBlock", pageBlock);
        model.put("pageSize", pageSize);

        . . .지난 경매도 map에 추가.

        return new ModelAndView("logon/myAuctionList", model);
    }
    else if(mode.equals("bid")) //mode=="bid" //내가
입찰한 경매에 대한 요청일 경우
        . . . //판매자일 경우와 마찬가지로 입찰 목록을 조회하고 게시판
형식으로 표시
        else
        {
            return new ModelAndView("default");
        }
    }

}
}

```

## Ajax관련

경매 정보(남은 시간을 제외한 다른 정보) 새로고침 없이 실시간으로 변경

```

function getOtherValues()
{
    var params = "";
    var i;
    for(i=0; i<pageSize; i++)
    {
        if(document.getElementById("itemnum"+i) != null)
        {
            //alert();
            params =
"itemnum="+document.getElementById("itemnum"+i).value;

```

```

        otherindex=i;
        request2 = new Request(setOtherValues,
"getOtherValue.do" , params , "GET" );
        request2.sendRequest();
    }
}

```

getOtherValues()로 sendRequest()하면 getOtherValue.do에서 처리를 하고  
setOtherValues에서 받은 값을 가지고 DOM 처리 innerHTML을 바꿔서 정보 변경하는 구조

```

function setOtherValues()
{
    //alert('setOtherValues');

    var msg="";
    var kingBidElement=document.getElementById("king"+otherindex);
    var readCountElement=document.getElementById("readcount"+otherindex);
    var bidCountElement=document.getElementById("bidcount"+otherindex);
    if( request2.httpRequest.readyState == 4 ){
        if( request2.httpRequest.status ==200 ){
            msg = request2.httpRequest.responseText;
            //alert(msg);
            var resArr=msg.split(" ");
            //alert(resArr);

kingBidElement.innerHTML=resArr[0]+"<br>입찰금:"+resArr[1]+"gold";
            readCountElement.innerHTML=resArr[2];
            bidCountElement.innerHTML=resArr[3];
        }
    }
}

```