

# NEEDS MARKET

-Used Market Platform-

**OBT Team** - 안수민, 김진수, 최지현



# CONTENTS



01

## OUR COMPANY

- 개발 동기 및 목표
- 샘플 페이지
- 개발 환경

02

## FUNCTION

- 메인페이지
- 로그인
- 회원관리
- 게시판
- 댓글
- 이미지

03

## SUMMARY

- 요약
- Q & A

# OUR GOAL



## MISSION

SPRING을 토대로  
소통형 홈페이지 구현



## VISION

중고나라를 베이스로 한  
중고상품 거래 플랫폼



## VALUES

게시판을 활용하여  
회원들이 거래 활동을  
할 수 있게 제작

# SAMPLE PAGE

중고나라

카카오톡만 유도, 택배거래만 요구, 안전결제 외부링크 유도시 거래금지

(주)중고나라는 통신판매중개자입니다. 판매 혹은 구매 회원 간의 상품거래정보 및 거래에 관여하지 않으며, 어떠한 의무와 책임도 부담하지 않습니다.

공지사항 · 이용정책 · 사기예방법 · 중고나라 ★ 이벤트 · 중고나라 ★ 공동구매

▶우리 동네 강원/제주

중고폰/모바일

- 모바일 협력사 신청
- 알뜰폰 유심요금제 가입
- 중나 모바일 가맹점 소개
- 중나 직매입 중고폰 시세
- ★100일AS보장&A급판매
- ★인증 중고폰 사고팔기
- 휴대폰 판매/매입
- 태블릿 판매/매입
- 주변기기/악세사리

브랜드 리미티드 리셀

- ★리셀 안전결제전용
- 조던&나이키
- 이치부스트&아디다스
- 기타 한정판

컴퓨터

- 컴퓨터 협력사 신청
- ★컴퓨터 안전결제전용
- 컴퓨터 자유게시판
- 컴퓨터 협력업체









중고나라

(주)중고나라는 통신판매중개자일 뿐이며, 통신판매자의 당사자가 아닙니다.

판매 회원과 구매 회원 간의 상품거래정보 및 거래에 관여하지 않으며, 어떠한 의무와 책임도 부담하지 않습니다.  
Copyright © Joonggonara All Rights Reserved.

재고 급처분

더보기 >

 <p>새도같은 리퍼 본사직영 쿠쿠 밀렉스 안마의자</p> <p>리싱크팀</p> <p>14:28 · 조회 59</p>	 <p>오도먼트 감속주행 레트로그램 배배로데이 에디션</p> <p>리싱크팀</p> <p>20.11.05. · 조회 473</p>	 <p>다이어트 패까지 제지방감소 녹차카테킨 오투에스</p> <p>리싱크팀</p> <p>20.11.02. · 조회 155</p>	 <p>신제품 반출상품 한정입고! 애플 에어팟 2세대</p> <p>리싱크팀</p> <p>20.10.29. · 조회 3,330</p>
 <p>초음파를 통한 감마선과 극자외선 Panasonic 초음파 리프터</p> <p>리싱크팀</p> <p>20.10.28. · 조회 333</p>	 <p>온도리얼 시즈닝&amp;연식물고기 증대 오세독세프의 착아일</p> <p>리싱크팀</p> <p>20.10.23. · 조회 901</p>	 <p>세제화된 포라피럼 용량 8배 방앤올름슨 기화진</p> <p>리싱크팀</p> <p>20.10.20. · 조회 4,682</p>	 <p>전용 5년 보장 완벽구상 삼성 갤럭시노트 101</p> <p>리싱크팀</p> <p>20.10.16. · 조회 5,982</p>

2015년 중고나라 나눔 공지

금...  
모은금 2,952,408  
참여수 506,790

저금하기 담아가기

사기피해사례 조회

전화, 계좌번호 검색

대치트는 범죄피해 방지를 위해  
2006년부터 운영되고 있습니다.  
최근 6개월내 1회 이상 등록된  
정보를 확인 할 수 있습니다.

제공: 대치트(TheCheat.co.kr)

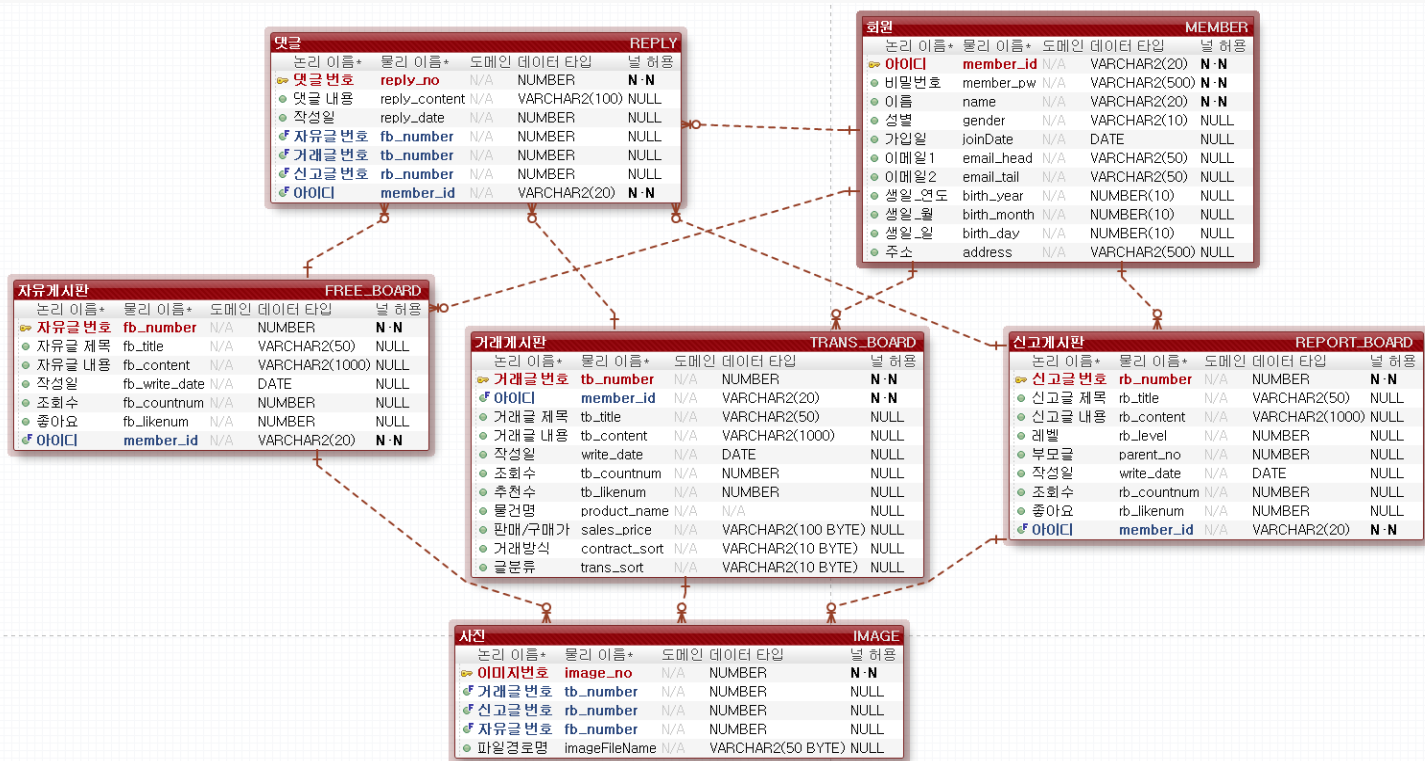
안전한 인터넷 거래  
경찰청 사이버안전국 협력사업  
전화번호 계좌번호  
검색

거래 전 반드시 상대방이 최근 3개월내 3회 이상 '사이버 사기'로 경찰청에 신고된 사실이 있는지 확인해 보세요  
예방수칙 신고하기

# 개발 환경

- ERD
- ECLIPSE - STS
- ORACLE
- Apache-Tomcat-9.0.46
- Git Hub

# ERD



# 메인 페이지



메인페이지



로그인



회원관리



게시판



이미지



댓글

```
</head>
<body>
  <div id="container">
    <div id="header">
      <tiles:insertAttribute name="header"/>
    </div>
    <div id="sidebar-left">
      <tiles:insertAttribute name="side"/>
    </div>
    <div id="content">
      <tiles:insertAttribute name="body"/>
    </div>
    <div id="footer">
      <tiles:insertAttribute name="footer"/>
    </div>
  </div>
</body>
```

# 로그인 페이지



## (1) 회원가입

```
@RequestMapping(value = "/addMember.do", method = RequestMethod.POST)
public ResponseEntity addMember(@ModelAttribute("memberVO") MemberVO _memberVO, HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    response.setContentType("text/html; charset=UTF-8");
    request.setCharacterEncoding("utf-8");
    String message = null;
    ResponseEntity resEntity = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");

    String password = _memberVO.getMember_pw();
    String encPassword = bCryptPasswordEncoder.encode(password);
    _memberVO.setMember_pw(encPassword);

    try {
        memberService.addMember(_memberVO);
        message = "<script>";
        message += " alert('회원 가입을 마쳤습니다. 로그인창으로 이동합니다.');" ;
        message += " location.href='" + request.getContextPath() + "/member/loginForm.do';";
        message += " </script>";
    } catch (Exception e) {
        message = "<script>";
        message += " alert('작업 중 오류가 발생했습니다. 다시 시도해 주세요');" ;
        message += " location.href='" + request.getContextPath() + "/member/memberForm.do';";
        message += " </script>";
        e.printStackTrace();
    }
    resEntity = new ResponseEntity(message, responseHeaders, HttpStatus.OK);
    return resEntity;
}
```



# 로그인 페이지

 메인페이지  로그인  회원관리  게시판  이미지  댓글

## (2) 중복 아이디

```
<select id="selectOverlappedID" parameterType="String"
resultType="String">
  select decode(count(*),1, 'true', 0, 'false')
  from member
  where member_id = #{id}
</select>
```

```
function fn_overlapped(){
  var member_id=$("#member_id").val();
  if(member_id==''){
    alert("ID를 입력하세요");
    return;
  }
  $.ajax({
    type:"post",
    async:false,
    url:"${contextPath}/member/overlapped.do",
    dataType:"text",
    data: {id:member_id},
    success:function (data,textStatus){
      if(data=='false'){
        alert("사용할 수 있는 ID입니다.");
        $('#btnOverlapped');
        $('#member_id');
        $('#_member_id').val(member_id);
      }else{
        alert("사용할 수 없는 ID입니다.");
      }
    },
    error:function(data,textStatus){
      alert("에러가 발생했습니다.");
    }
  });
}
```

# 로그인 페이지



메인페이지



로그인



회원관리



게시판



이미지



댓글

## (3) 로그인

```
@RequestMapping(value = "/standard_login.do", method = { RequestMethod.POST, RequestMethod.GET })
public ModelAndView login(@ModelAttribute("memberVO") MemberVO memberVO, HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    ModelAndView mav = new ModelAndView();

    MemberVO login = memberService.standard_login(memberVO);

    if (login != null && bCryptPasswordEncoder.matches(memberVO.getMember_pw(), login.getMember_pw())) {
        HttpSession session = request.getSession();
        session.setAttribute("memberInfo", login);
        session.setAttribute("isLogin", true);

        String action = (String) session.getAttribute("action");
        if (action != null && action.equals("/main.do")) {
            mav.setViewName("forward:" + action);
        } else {
            mav.setViewName("redirect:/main.do");
        }
    } else {
        String wronglogin="잘못된 로그인";
        mav.addObject("wronglogin", wronglogin);
        mav.setViewName("/member/loginForm");
    }

    return mav;
}
```

# 로그인 페이지

  
메인페이지

  
로그인

  
회원관리

  
게시판

  
이미지

  
댓글

## (4) 로그아웃

```
@RequestMapping(value = "/logout.do", method = RequestMethod.GET)
public ModelAndView logout(HttpServletRequest request, HttpServletResponse response) throws Exception {
    HttpSession session = request.getSession();
    session.setAttribute("isLogin", false);
    session.removeAttribute("memberInfo");
    mav.setViewName("redirect:/main.do");
    return mav;
}
```

# 회원관리 페이지



## 메인페이지



로그인



## 회원관리



게시판



## 이미지



대그슬

## (1) 회원 수정

```
@RequestMapping(value="/modmemberInfo.do" ,method = RequestMethod.POST)
@ResponseBody
public ResponseEntity modmemberInfo(HttpServletRequest request,
    HttpServletResponse response) throws Exception{
    request.setCharacterEncoding("utf-8");
    Map memberInfoMap = new HashMap();
    Enumeration enu=request.getParameterNames();
    while(enu.hasMoreElements()){
        String name=(String)enu.nextElement();
        String value=request.getParameter(name);
        memberInfoMap.put(name,value);
    }
    System.out.println("memberInfoMap>>>>>>>"+memberInfoMap);
    String id = memberVO.getMember_id();
    memberInfoMap.put("id", id);

    String message;
    ResponseEntity resEnt=null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        memberService.modmember(memberInfoMap);
        message = "<script>";
        message += " alert('회원정보를 수정했습니다.')";
        message += " location.href='"+request.getContextPath()+"/main.do'";
        message += "</script>";
        resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
    }
```

# 회원관리 페이지



## (2) 회원 탈퇴

```
@RequestMapping(value = "/removeMember.do", method = RequestMethod.POST)
public ResponseEntity memberDelete(MemberVO memberVO, HttpServletRequest request, HttpServletResponse response,
    HttpSession session, RedirectAttributes rtrr) throws Exception {
    memberVO = (MemberVO) session.getAttribute("memberInfo");
    String sessionPass = memberVO.getMember_pw();
    String voPass = memberVO.getMember_pw();
    String message;
    ResponseEntity resEnt = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        if (sessionPass.equals(voPass)) {
            memberService.removeMember(memberVO);
            session.invalidate();
            message = "<script>";
            message += " alert('탈퇴에 성공했습니다. 다음에 또 봐요!');";
            message += " location.href='" + request.getContextPath() + "/member/loginForm.do'";
            message += " </script>";
            resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
        }
    } catch (Exception e) {
        message = "<script>";
        message += " alert('작업중 오류가 발생했습니다. 다시 시도해 주세요.')";
        message += " location.href='" + request.getContextPath() + "/member/memberInfo.do'";
        message += " </script>";
        resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
        e.printStackTrace();
    }
    return resEnt;
}
```



# 이미지 업로드

 메인페이지

 로그인

 회원관리

 게시판

 이미지

 댓글

```
function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function(e) {
            $(input).parent().next().find('.preview').attr('src',
                e.target.result);
        }
        reader.readAsDataURL(input.files[0]);
    }
}

function backToList(obj) {
    obj.action = "${contextPath}/board/rb_board/report_boardlist.do";
    obj.submit();
}

var cnt = 0;
function addFile() {
    cnt++;
    $("#file-list")
        .append(
            "<tr><td><input type='file' name='file'
                + cnt
                + "' onChange='readURL(this);' /><a href='#this' name='file-delete'>삭제</a></td><td><img class='preview' id=file"+cnt
        );
    $(".a[name='file-delete']").on("click", function(e) {
        e.preventDefault();
        console.log("$this", $(this));
        deleteFile($(this));
    });
}
```

# 이미지 업로드



```
// 다중 이미지 글쓰기
@RequestMapping(value = "/tb_board/addNewtrans.do", method = { RequestMethod.GET, RequestMethod.POST })
@ResponseBody
public ResponseEntity addNewArticle(MultipartHttpServletRequest multipartRequest, HttpServletResponse response)
    throws Exception {
    multipartRequest.setCharacterEncoding("utf-8");
    String imageFileName = null;
    Map articleMap = new HashMap();
    Enumeration enu=multipartRequest.getParameterNames();
    while (enu.hasMoreElements()) {
        String name = (String) enu.nextElement();
        String value = multipartRequest.getParameter(name);
        articleMap.put(name, value);
    }
}
```

# 이미지 업로드



메인페이지



로그인



회원관리



게시판



이미지



댓글

```
// 로그인 시 세션에 저장된 회원 정보에서 글쓴이 아이디를 얻어와서 Map에 저장합니다.
HttpSession session = multipartRequest.getSession();
MemberVO memberVO = (MemberVO) session.getAttribute("memberInfo");
if (memberVO != null && memberVO.getMember_id() != null) {
    String id = memberVO.getMember_id();
    articleMap.put("member_id", id);
    articleMap.put("imageFileName", imageFileName);
}
System.out.println("upload Controller >>> " + articleMap);
List<String> fileList = upload(multipartRequest);
List<ImageVO> imageFileList = new ArrayList<ImageVO>();
if (fileList != null && fileList.size() != 0) {
    for (String fileName : fileList) {
        ImageVO imageVO = new ImageVO();
        imageVO.setImageFileName(fileName);
        imageFileList.add(imageVO);
        System.out.println("컨트롤러 fileList>>>>>>>>>" + fileName);
    }
    articleMap.put("imageFileList", imageFileList);
}
String message;
ResponseEntity resEnt = null;
HttpHeaders responseHeaders = new HttpHeaders();
responseHeaders.add("Content-Type", "text/html; charset=utf-8");
```



# 이미지 업로드



메인페이지



로그인



회원관리



게시판



이미지



댓글

```
// 다중 이미지 업로드하기
```

```
private List<String> upload(MultipartHttpServletRequest multipartRequest) throws Exception{
    List<String> fileList= new ArrayList<String>();
    Iterator<String> fileNames = multipartRequest.getFileNames();
    while(fileNames.hasNext()){
        String fileName = fileNames.next();
        MultipartFile mFile = multipartRequest.getFile(fileName);
        String originalFileName=mFile.getOriginalFilename();
        UUID uuid = UUID.randomUUID();
        originalFileName = uuid.toString() + "_" + originalFileName;
        fileList.add(originalFileName);
        File file = new File(ARTICLE_IMAGE_REPO + "\\\" + fileName);
        if(mFile.getSize()!=0){ //File Null Check
            if(! file.exists()){
                if(file.getParentFile().mkdirs()){
                    file.createNewFile();
                }
            }
            mFile.transferTo(new File(ARTICLE_IMAGE_REPO + "\\\" + "temp" + "\\\" + originalFileName));
        }
    }
    return fileList;
}
```

# 게시판 특징

  
메인페이지

  
로그인

  
회원관리

  
게시판

  
이미지

  
댓글

## TRANS BOARD

판매

글 쓰기  
가격 제시  
이미지 업로드 필수  
댓글 가능

구매

글 쓰기  
원하는 가격 제시  
이미지 업로드 옵션  
댓글 가능

## FREE BOARD

자유롭게 회원들이 소통하는 공간

글 쓰기  
이미지 업로드  
댓글

## REPORT BOARD

거래 중에 발생한 사기나  
경험담을 공유하는 공간

글 쓰기  
이미지 업로드  
댓글  
답글

# 게시판 페이지



## 답글(1)

```
// 답글쓰기 창 넘어가기
@RequestMapping(value = "/rb_board/rb_rearticleForm.do", method = { RequestMethod.GET, RequestMethod.POST })
private ModelAndView reArticleForm(HttpServletRequest request, HttpServletResponse response) throws Exception {
    String viewName = (String) request.getAttribute("viewName");
    int parent_no = Integer.parseInt(request.getParameter("parent_no"));
    ModelAndView mav = new ModelAndView();
    mav.addObject("parent_no", parent_no);
    mav.setViewName(viewName);
    return mav;
}
```

```
// 게시판 목록 보여주기
```

## 답글(2)

```
// 답글 쓰기
@RequestMapping(value = "/rb_board/report_boardNewReArticle.do", method = { RequestMethod.GET, RequestMethod.POST })
public ResponseEntity addNewReply(HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, Object> replyMap = new HashMap<String, Object>();
    Enumeration<String> enu = request.getParameterNames();
    while (enu.hasMoreElements()) {
        String name = (String) enu.nextElement();
        String value = request.getParameter(name);
        replyMap.put(name, value);
    }

    System.out.println("replyMap>>>>>" + replyMap);
    String message;
    ResponseEntity resEnt = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        report_boardservice.insertNewreply(replyMap);
        message = "<script>";
        message += " alert('답글을 추가했습니다.');"
        message += " location.href='" + request.getContextPath() + "/board/rb_board/rb_listarticles.do' ";
        message += " </script>";
        resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
    } catch (Exception e) {
    }
}
```

# 게시판 페이지



## 글 수정(1)

```
<input type=button value="수정하기" class="btn btn-info btn-sm" 이 버튼을 통해 수정버튼으로 이동
  onClick="fn_modify_article('${contextPath}/board/rb_board/rb_articlemodForm.do', ${article.rb_num
<input type=button value="삭제하기" class="btn btn-danger btn-sm"
  onClick="fn_remove_article('${contextPath}/board/rb_board/removeReport_board.do' , ${article.rb_n
/c:if>
input type=button value="추천하기" class="btn btn-warning btn-sm"
  onClick="fn_recommend_article('${contextPath}/board/rb_board/recommend.do' , ${article.rb_numb
input type=button value="답글쓰기" class="btn btn-danger btn-sm" onClick="fn_reply_form('${contextPat
input type=button value="리스트로 돌아가기" class="btn btn-primary btn-sm" onClick="backToList(this.form
|>
```

# 게시판 페이지



## 글 수정(2)

```
3     ModelAndView mav = new ModelAndView();
4     mav.setViewName(viewName);
5     return mav;
6 }
7
8 // 수정하기 창 넘어가기           수정창으로 이동
9 • @RequestMapping(value = "/rb_board/rb_articlemodForm.do", method = { RequestMethod.GET, RequestM
10 private ModelAndView modArticleform(HttpServletRequest request, HttpServletResponse response) throws Ex
11     String viewName = (String) request.getAttribute("viewName");
12     int rb_number = Integer.parseInt(request.getParameter("rb_number"));
13     ModelAndView mav = new ModelAndView();
14     mav.addObject("rb_number", rb_number);
15     mav.setViewName(viewName);
16     return mav;
17 }
18
19 // 답글쓰기 창 넘어가기
20 • @RequestMapping(value = "/rb_board/rb_rearticleForm.do", method = { RequestMethod.GET, RequestM
21 private ModelAndView reArticleform(HttpServletRequest request, HttpServletResponse response) throws Ex
22     String viewName = (String) request.getAttribute("viewName");
```

# 게시판 페이지



## 글 삭제(1)

```
</c:if>
<tr id="tr_btn">
  <td colspan="2" align="center"><c:if
    test="${memberInfo.member_id == article.member_id}">
    <input type="button" value="수정하기" class="btn btn-info btn-sm"
      onClick="fn_modify_article('${contextPath}/board/rb_board/rb_articlemodForm.do', ${arti
    <input type="button" value="삭제하기" class="btn btn-danger btn-sm"
      onClick="fn_remove_article('${contextPath}/board/rb_board/removeReport_board.do', ${
    </c:if>
    <input type="button" value="추천하기" class="btn btn-warning btn-sm"
      onClick="fn_recommend_article('${contextPath}/board/rb_board/recommend.do', ${artic
    <input type="button" value="답글쓰기" class="btn btn-danger btn-sm" onClick="fn_reply_form('${
    <input type="button" value="리스트로 돌아가기" class="btn btn-primary btn-sm" onClick="backToLi
  </td>
</tr>
</table>
</form>
-- 댓글 -->
<c:set var="reply" value="${resultMap.reply}" />
</script>
```

# 게시판 페이지



## 글 삭제(2)

```
// 글 삭제하기
@RequestMapping(value = "/rb_board/removearticle.do", method = { RequestMethod.POST })
@ResponseBody
public ResponseEntity removeArticle(@RequestParam("rb_number") int rb_number, HttpServletRequest req,
    HttpServletResponse response) throws Exception {
    response.setContentType("text/html; charset=UTF-8");
    String message;
    ResponseEntity resEnt = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        report_boardservice.removeArticle(rb_number);
        message = "<script>";
        message += " alert('글을 삭제했습니다.');"";
        message += " location.href=" + request.getContextPath() + "/board/rb_board/rb_listarticles.do";
        message += " </script>";
        resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
    } catch (Exception e) {
        message = "<script>";
        message += " alert('작업중 오류가 발생했습니다.다시 시도해 주세요.');"";
        message += " location.href=" + request.getContextPath() + "/board/rb_board/rb_listarticles.do";
        message += " </script>";
    }
}
```



# 게시판 페이지

 메인페이지

 로그인

 회원관리

 게시판

 이미지

 댓글

## 글 추천(1)

```
update REPORT_BOARD
set
rb_title=#{rb_title}, rb_content=#{rb_content}
where
rb_number=#{rb_number}
</update>

<update id="recommend" parameterType="int">
update REPORT_BOARD
set rb_likenum = rb_likenum+1
where rb_number = #{rb_number}
</update>

</mapper>
```

in Source

# 게시판 페이지



## 글 추천(2)

```
rb_View.jsp Report_BoardController.java x
99     return mav;
100 }
101
102 // 추천 기능
103 @RequestMapping(value = "/rb_board/recommend.do", method = { RequestMethod.GET, RequestMethod.POST })
104 public ResponseEntity recommend(HttpServletRequest request, HttpServletResponse response) throws Exception {
105     response.setContentType("text/html; charset=UTF-8");
106     String message;
107     ResponseEntity resEnt = null;
108     HttpHeaders responseHeaders = new HttpHeaders();
109     responseHeaders.add("Content-Type", "text/html; charset=utf-8");
110     int rb_number = Integer.parseInt(request.getParameter("rb_number"));
111     System.out.println("컨트롤러 타는거야???>>>" + rb_number);
112
113     try {
114         report_boardservice.recommend(rb_number);
115         message = "<script>";
116         message += " alert('추천을 완료했습니다.');" + request.getContextPath() + "/board/rb_board/rb_listarticles.do';";
117         message += "</script>";
118         resEnt = new ResponseEntity(message, responseHeaders, HttpStatus.CREATED);
119     } catch (Exception e) {
120         message = "<script>";
121         message += " alert('작업중 오류가 발생했습니다. 다시 시도해 주세요.');" + request.getContextPath() + "/board/rb_board/rb_listarticles.do';";
122     }
123     return resEnt;
124 }
```

History Synchronize Git Reflog Search Servers Git Staging Console x

Tomcat v9.0 Server at localhost [Apache Tomcat] C:\JAVA\bin\javaw.exe (2020. 11. 9 오후 10:29:57)

# 게시판 페이지



메인페이지



로그인



회원관리



게시판



이미지



댓글

## 글 조회수(1)

```
57  
58     @Override  
59     public Map<String, Object> viewArticle(int rb_number) throws Exception {  
60         Report_boardVO report_boardvo = report_Boarddao.selectArticle(rb_number);  
61         Map<String, Object> resultMap = new HashMap<String, Object>();  
62         resultMap.put("reportBoard", report_boardvo);  
63         String boardKind = "rb";  
64         List<ReplyVO> replList = replyDAO.selectAllReplyList(rb_number, boardKind);  
65         resultMap.put("replList", replList);  
66         List<ImageVO> imageFileList = report_Boarddao.selectImageFileList(rb_number);  
67         resultMap.put("imageFileList", imageFileList);  
68         report_Boarddao.boardHit(rb_number);  
69         return resultMap;  
70     }  
71
```

# 게시판 페이지



메인페이지



로그인



회원관리



게시판



이미지



댓글

## 글 조회수(1)

```
4 </delete>
5
6 <select id="selectNewArticleNO" resultType="int">
7   <![CDATA[
8     SELECT nvl(max(rb_number), 0) + 1 from REPORT_BOARD
9   ]]>
10 </select>
11
12 <update id="boardHit" parameterType="int">
13   UPDATE REPORT_BOARD SET
14   rb_countnum = rb_countnum+1
15   WHERE rb_number = #{rb_number}
16 </update>
17
18
19
20 <insert id="insertNewImage" parameterType="java.util.Map">
21   <foreach item="item" collection="list" open="INSERT ALL"
22     separator=" " close="SELECT
23     * FROM DUAL"> INTO
```

# 게시판 페이지



메인페이지



로그인



회원관리



게시판



이미지



댓글

## 글목록 보기

```
// 게시판 목록 보여주기
@RequestMapping(value = "/fb_board/fb_listArticles.do", method = { RequestMethod.GET, RequestMethod.POST })
public ModelAndView listArticles(HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    String viewName = (String) request.getAttribute("viewName");
    List<FreeBoardVO> articlesList = freeBoardService.listArticles();
    ModelAndView mav = new ModelAndView(viewName);
    mav.addObject("articlesList", articlesList);
    return mav;
}
```

[illegible]

## 댓글목록 (2)

```

@Transactional(isolation = Isolation.READ_COMMITTED)
@Override
public Map<String, Object> viewArticle(int fb_number) throws Exception {
    FreeBoardVO freeBoardVO = freeBoardDAO.selectArticle(fb_number);
    Map<String, Object> resultMap = new HashMap<String, Object>();
    resultMap.put("freeBoard", freeBoardVO); VO를 맵에 넣어줘 제 선언을 방지
    String boardKind = "fb";
    List<ReplyVO> replList = replyDAO.selectAllReplyList(fb_number, boardKind);
    resultMap.put("replList", replList); ReplyDao 부르기 / boardKind로 게시판 분류
    List<ImageVO> imageFileList = freeBoardDAO.selectImageFileList(fb_number);
    resultMap.put("imageFileList", imageFileList);
    freeBoardDAO.boardHit(fb_number);
    return resultMap;
}
    
```



## 댓글목록 (3)

```
public List<ReplyVO> selectAllReplyList(int number, String boardKind) {  
    if (boardKind.equals("fb")) {  
        return sqlSession.selectList("mapper.reply.select_fbReply", number);  
    } else if (boardKind.equals("tb")) {  
        return sqlSession.selectList("mapper.reply.select_tbReply", number);  
    } else {  
        return sqlSession.selectList("mapper.reply.select_rbReply", number);  
    }  
}
```



## 댓글 등록 (1)

```

<c:set var="reply" value="${resultMap.reply}" />
<script
    src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.0/css/solid.min.css"></script>
<form id="replyTable" action="${contextPath}/reply/addNewReply.do"
    method="post" onsubmit="return loginCheck();" >
    Submit 전에 자바 스크립트 실행 체크

    /* 로그인 체크 */
    function loginCheck() {
        if($('#member_id').val() == "") {
            alert("로그인을 해주세요!");
            location.href = "${contextPath}/member/loginForm.do";
            return false;
        } else {
            return true;
        }
    }
    }>
    필요한 데이터
    hidden으로 넘기기

    <input type="hidden" name="boardtype" value="free_board" /></td>
    <td><textarea name="reply_content" rows="3" cols="100"
        maxlength="500" placeholder="댓글을 남겨보세요"></textarea></td>
    <td><button class="addRepl" type="submit">댓글 달기</button> </td>
</tr>
    
```

## 댓글 등록 (2)

```

@RequestMapping(value = "/reply/addNewReply.do", method = { RequestMethod.POST, RequestMethod.GET })
public ModelAndView addNewReply(@ModelAttribute("replyVO") ReplyVO replyVO,
    @RequestParam("boardtype") String boardtype, @RequestParam("article_number") String article_number,
    HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    request.setCharacterEncoding("utf-8");
    //boardtype을 이용해 해당 글번호 산출
    String viewName;
    if (boardtype.equals("free_board")) {
        replyVO.setFb_number(article_number);
        viewName = "redirect:/board/fb_board/fb_View.do?fb_number=" + article_number;
    } else if (boardtype.equals("trans_board")) {
        replyVO.setTb_number(article_number);
        viewName = "redirect:/board/tb_board/tb_View.do?tb_number=" + article_number;
    } else {
        replyVO.setRb_number(article_number);
        viewName = "redirect:/board/rb_board/rb_View.do?rb_number=" + article_number;
    }
    ModelAndView mav = new ModelAndView(viewName);
    replyService.addNewReply(replyVO);
    return mav;
}
    
```

## 댓글 등록 (1)

```
public int addNewReply(ReplyVO replyVO) {  
    int reply_no = selectNewReplyNO();  
    replyVO.setReply no(reply no);  
    int result = sqlSession.insert("mapper.reply.insertNewReply", replyVO);  
    return result;  
}
```

```
<insert id="insertNewReply" parameterType="replyVO">  
    <![CDATA[  
        INSERT into reply(reply_no, reply_content, reply_date, fb_number, tb_number, rb_number, member_id)  
        VALUES(#{reply_no}, #{reply_content}, sysdate, #{fb_number, jdbcType=INTEGER},  
        #{tb_number, jdbcType=INTEGER}, #{rb_number, jdbcType=INTEGER}, #{member_id})  
    ]]>  
    값이 없으면 null로 넣기  
</insert>
```

## 댓글 삭제 (1)

```
/* ---댓글 삭제 ajax */
$(document).ready(function(){
    $(".delRepl").click(function(e){
        e.preventDefault();
        var reply_no = $(this).attr("id");
        var param = JSON.stringify({"boardKind": "fb", "reply_no": reply_no});

        $.ajax({
            method: "POST",
            url: "${contextPath}/reply/removeReply.do",
            contentType: "application/json; charset=UTF-8",
            data: param,
            success: function (res) {
                if(res=='success'){
                    $('.'+reply_no).remove();
                }
            },
            error: function (status) {
            }
        });
    });
});
```

## 댓글 삭제 (2)

```
@RequestMapping(value = "/reply/removeReply.do", method = RequestMethod.POST)
public @ResponseBody ResponseEntity removeReply(@RequestBody Map<String, String> replyMap,
    HttpServletResponse response) throws Exception {
    int reply_no = Integer.parseInt((String) replyMap.get("reply_no"));
    ResponseEntity resEnt;
    try {
        replyService.deleteReply(reply_no);
        resEnt = new ResponseEntity("success", HttpStatus.OK);
    } catch (Exception e) {
        resEnt = new ResponseEntity("fail", HttpStatus.NOT_MODIFIED);
        e.printStackTrace();
    }
    return resEnt;
}
```

**Thank  
You**

