

게시판 - 댓글 시스템 -

댓글 시스템 – 서버

❖ 댓글 테이블 정의

```
CREATE TABLE REPLIES(  
    REPLY_ID          NUMBER PRIMARY KEY,  
    BOARD_ID          NUMBER,  
    REPLY_LEVEL       NUMBER,  
    PARENT            NUMBER,  
    WRITER            VARCHAR2(20) NOT NULL,  
    CONTENT            VARCHAR2(1024),  
    DELETED           NUMBER(1),  
    REG_DATE          DATE,  
    UPDATE_DATE       DATE,  
    CONSTRAINT F_REPLIES_BOARD FOREIGN KEY(BOARD_ID)  
        REFERENCES BOARDS(BOARD_ID),  
    CONSTRAINT F_REPLIES_MEMBER FOREIGN KEY(WRITER)  
        REFERENCES MEMBERS(USER_ID)  
);
```

❖ 시퀀스 정의

```
CREATE SEQUENCE REPLIES_SEQ;
```

댓글 시스템 – 서버

❖ 모델 : Reply

```
@Data
public class Reply {
    private int    replyId;    // 댓글 번호
    private int    boardId;    // 글 그룹(게시글 ID)
    private int    replyLevel; // 댓글 수준
    private int    parent;     // 상위 글 번호
    private String writer;     // 작성자 ID
    private String content;    // 내용
    private int    deleted;    // 삭제여부
    private Date   regDate;
    private Date   updateDate;
}
```

댓글 시스템 – 서버

❖ Mapper : reply-maaper.xml

```
<mapper namespace="edu.iot.butter.dao.ReplyDao">
```

```
    <select id="getCount" parameterType="int" resultType="int">
        SELECT count(*) FROM replies
        WHERE board_id=#{boardId}
    </select>
```

```
    <select id="selectOne" parameterType="int" resultType="Reply">
        SELECT * FROM replies
        WHERE reply_id=#{replyId}
    </select>
```

```
    <select id="selectList" parameterType="int" resultType="Reply">
        SELECT * FROM replies
        WHERE board_id=#{boardId}
        ORDER BY reply_level, reg_date desc
    </select>
```

댓글 시스템 – 서버

❖ Mapper : reply-maaper.xml

```
<insert id="insert" parameterType="Reply" useGeneratedKeys="true"
      keyColumn="reply_id" keyProperty="replyId">
    insert into replies
      (reply_id, board_id, reply_level, parent,
       writer,deleted, content, reg_date, update_date)
    values(REPLIES_SEQ.NEXTVAL, #{boardId}, #{replyLevel},
          #{parent}, #{writer}, 0, #{content}, sysdate, sysdate)
</insert>

<update id="update" parameterType="Reply">
    update replies set
      content = #{content},
      update_date = sysdate
    WHERE reply_id=#{replyId}
</update>
```

댓글 시스템 – 서버

❖ Mapper : reply-maaper.xml

```
<update id="delete" parameterType="int">
    update replies set
        content = '삭제된 글입니다.',
        deleted = 1,
        update_date = sysdate
    WHERE reply_id=#{replyId}

</update>
</mapper>
```

댓글 시스템 – 서버

❖ DAO 인터페이스 : ReplyDao

```
public interface ReplyDao {  
    int getCount(int boardId) throws Exception;  
  
    Reply selectOne(int replyId) throws Exception;  
  
    List<Reply> selectList(int boardId) throws Exception;  
  
    int insert(Reply reply) throws Exception;  
  
    int update(Reply reply) throws Exception;  
  
    int delete(int replyId) throws Exception;  
}
```

댓글 시스템 – 서버

❖ 서비스 인터페이스 : ReplyService

```
public interface ReplyService {  
    List<Reply> getList(int boardId) throws Exception;  
  
    Reply get(int replyId) throws Exception;  
  
    Reply create(Reply reply) throws Exception;  
  
    Reply update(Reply reply) throws Exception;  
  
    boolean delete(int replyId) throws Exception;  
  
}
```


댓글 시스템 – 서버

❖ 서비스 인터페이스 구현 : ReplyServiceImpl

```
@Service
public class ReplyServiceImpl implements ReplyService {
    @Autowired
    ReplyDao dao;

    @Override
    public List<Reply> getList(int boardId) throws Exception {
        return dao.selectList(boardId);
    }

    @Override
    public Reply get(int replyId) throws Exception {
        return dao.selectOne(replyId);
    }
}
```

댓글 시스템 – 서버

❖ 서비스 인터페이스 구현 : ReplyServiceImpl

```
@Transactional
@Override
public Reply create(Reply reply) throws Exception {
    dao.insert(reply);
    return dao.selectOne(reply.getReplyId());
}

@Transactional
@Override
public Reply update(Reply reply) throws Exception {
    dao.update(reply);
    return dao.selectOne(reply.getReplyId());
}

@Transactional
@Override
public boolean delete(int replyId) throws Exception {
    return dao.delete(replyId)==1;
}
}
```

❖ REST 컨트롤러 : ReplyApiController

```
@RestController
@RequestMapping("/api/reply/{boardId}")
public class ReplyApiController {
    @Autowired
    ReplyService service;

    @RequestMapping(method = RequestMethod.GET)
    public List<Reply> getList(@PathVariable int boardId)
        throws Exception {
        return service.getList(boardId);
    }

    @RequestMapping(value =("/{replyId}", method = RequestMethod.GET)
    public Reply get(@PathVariable int boardId,
                    @PathVariable int replyId) throws Exception{
        return service.get(replyId);
    }
}
```

❖ REST 컨트롤러 : ReplyApiController

```
@RequestMapping(method = RequestMethod.POST)
public Reply insert(@PathVariable int boardId,
                   @RequestBody Reply reply) throws Exception{
    return service.create(reply);
}

@RequestMapping(method = RequestMethod.PUT)
public Reply update(@PathVariable int boardId,
                   @RequestBody Reply reply) throws Exception{
    return service.update(reply);
}

@RequestMapping(value =("/{replyId}",
                      method = RequestMethod.DELETE)
public boolean delete(@PathVariable int boardId,
                     @PathVariable int replyId) throws Exception{
    return service.delete(replyId);
}
}
```

댓글 시스템 - 뷰

❖ 뷰 기본 골격 : view.jsp

```
<script>
$(function() {

});
</script>
:
```

```
<div>
    ${board.content}
</div>
```

```
<c:if test="${not empty USER.userId}">
    <div id="reply-board"></div>
</c:if>
```

댓글 쓰기 영역 :
로그인 사용자만 활성화

```
<div id="reply-list"></div>
:
```

댓글 목록 영역

댓글 시스템 - 클라이언트

❖ Ajax 지원 클래스 정의 : /resources/js/rest.js

```
class Rest {  
    constructor(url) {  
        this.url = url;    // api URL  
    }  
  
    // 목록 얻기  
    list(params, callback) {  
        $.get(this.url + '?' + params, callback);  
    }  
  
    // 항목 하나 얻기  
    get(id, callback) {  
        $.get(this.url + '/' + id, callback);  
    }  
}
```

댓글 시스템 - 클라이언트

❖ Ajax 지원 클래스 정의 : /resources/js/rest.js

```
create(data, callback) {
    $.ajax({
        url : this.url,
        type : 'post',
        contentType : 'application/json',
        data : JSON.stringify(data),
        success : callback
    });
}

update(data, callback) {
    return $.ajax({
        url : this.url,
        type : 'put',
        contentType : 'application/json',
        data : JSON.stringify(data),
        success : callback
    });
}
```

댓글 시스템 - 클라이언트

❖ Ajax 지원 클래스 정의 : /resources/js/rest.js

```
remove(id, callback) {  
    return $.ajax({  
        url : this.url + '/' + id,  
        type : 'delete',  
        success : callback  
    });  
}  
}
```


댓글 시스템 - 클라이언트

❖ 뷰 : view.js - REST 객체 준비

```
<script src="<c:url value="/resources/js/rest.js"/>"></script>
<script>
<c:url value="/api/reply/${board.boardId}" var="apiUrl"/>
var api = new Rest('${apiUrl}');
$(function() {
    var opt = {
        api : api,                // Ajax 통신 지원 객체
        boardId : ${board.boardId}, // 현재 게시글
        writer : '${USER.userId}',  // 신규 댓글 작성자 ID
        replyList : $('#reply-list'); // 댓글 목록 엘리먼트
    };
});
</script>
```

댓글 시스템

- replyBoard 플러그인 -

replyBoard 플러그인

❖ 뷰 : view.js - 댓글 쓰기 플러그인

댓글 쓰기

등록

```
<div id="reply-board"> </div>
```

- 플러그인 이름 : replyBoard
- 적용 : \$('#reply-board').replyBoard(opt);

replyBoard 플러그인

❖ 댓글 관련 템플릿 객체: /resources/js/reply.templ.js

```
var replyTempl = {  
  replyAddTempl : `  
    <div class="card card-body p-2">  
      <h6 class="card-title">댓글 쓰기</h6>  
      <textarea id="reply-content" class="card-text"  
        rows=4"></textarea>  
      <div class="flex-row text-right">  
        <a id="reply-add"  
          class="card-link btn btn-primary btn-sm">  
            <i class="fa fa-share-square"></i> 등록</a>  
      </div>  
    </div>`,  
};
```

replyBoard 플러그인

❖ 댓글 쓰기 플러그인: /resources/js/reply.js

```
$.fn.replyBoard = function(opt, replyList) {
    var templ = $(replyTempl.replyAddTempl);
    var content = templ.find('#reply-content');
    templ.find('#reply-add').click(function(e){
        var reply = {
            boardId : opt.boardId,
            writer : opt.writer,
            parent : 0,
            replyLevel : 0,
            content : content.val()
        };
        api.create(reply, function(data) {
            data.regDate = new Date(data.regDate); // 문자열 -> Date 변환
            content.val(''); // <textarea> 내용 지우기
            // 댓글 리스트에 추가 ?
        });
    });
}

this.append(templ);
return this;
}
```

replyBoard 플러그인

❖ 뷰 : view.js - 댓글 쓰기 플러그인 적용

```
<script src="<c:url value="/resources/js/rest.js"/>"></script>
<script src="<c:url value="/resources/js/reply.templ.js"/>"></script>
<script src="<c:url value="/resources/js/reply.js"/>"></script>
<script>
<c:url value="/api/reply/${board.boardId}" var="apiUrl"/>
var api = new Rest('${apiUrl}');
$(function() {
    var opt = {
        api : api,                // REST API 객체
        boardId : ${board.boardId}, // 현재 게시글 ID
        writer : '${USER.userId}',  // 현재 사용자 ID
        replyList : $('#reply-list') // 댓글 목록 엘리먼트
    };

    $('#reply-board').replyBoard(opt);
});
</script>
```

replyBoard 플러그인

❖ 댓글 관련 템플릿 객체: /resources/js/reply.templ.js

```
var replyTempl = {  
  replyAddTempl : `...`,  
  
  // 한개의 Reply에 대응하는 Media Object 생성  
  mediaObjectTempl : function(reply, writer) {  
    var obj = $(`  
      <div class="media d-block d-md-flex mt-3"  
        data-reply-id="${reply.replyId}"  
        data-reply-level="${reply.replyLevel}">  
  
          
    `);  
  }  
};
```

replyBoard 플러그인

❖ 댓글 관련 템플릿 객체: /resources/js/reply.templ.js

```
<div class="media-body text-center
      text-md-left ml-md-3 ml-0">
  <div style="overflow:hidden">
    <h5 class="mt-0 font-weight-bold float-left">
      ${reply.writer}</h5>
    <div class="buttons float-right ml-3"></div>
    <div class="float-right">작성일:${reply.regDate}</div>
  </div>
  <div class="reply-content">${reply.content}</div>
  <div class="work"></div>
  <div class="children"></div>
</div>`);
// 로그인 사용자에게 따라 버튼의 종류 구분하여 추가
this.buttonsTempl(obj, reply, writer);
return obj;
},
```


replyBoard 플러그인

❖ 댓글 관련 템플릿 객체: /resources/js/reply.templ.js

```
buttonsTempl : function(item, reply, writer) {  
    // 삭제 글이 아닌 경우  
    if(!reply.deleted) {  
        // 로그인 사용자인경우 댓글 버튼 추가  
        if( writer != '' ) {  
            item.find('.buttons').append('<button class="reply-add-show"><i class="fa fa-reply"></i></button>');  
        }  
        // 자신이 작성한 댓글인경우 수정 버튼과 삭제 버튼 추가  
        if(reply.deleted == 0 && reply.writer == writer) {  
            item.find('.buttons').append(  
                '<button class="reply-edit-show"><i class="fa fa-edit"></i></button>'+  
                '<button class="reply-delete"><i class="fa fa-trash"></i></button>');  
        }  
    } else { // 삭제된 댓글  
        item.addClass('deleted-reply')  
    }  
}
```

replyBoard 플러그인

❖ 댓글 쓰기 플러그인: /resources/js/reply.js

```
$.fn.replyBoard = function(opt, replyList) {  
    :  
    api.create(reply, function(data) {  
        data.regDate = new Date(data.regDate); // 문자열 -> Date 변환  
        content.val(''); // <textarea> 내용 지우기  
        // 댓글 리스트에 추가  
        opt.replyList.prepend(  
            replyTempl.mediaObjectTempl(data, opt.writer));  
    });  
})  
  
this.append(templ);  
return this;  
}
```

댓글 시스템

- replyList 플러그인 -

replyList 플러그인

❖ 목록 보여주기



admin

abcd

작성일:2018-04-17



abc

abc가 댓글을 합니다.

작성일:2018-04-17



admin

댓글 첫번째 테스트2

작성일:2018-04-17



replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
// 1차원 배열을 트리 구조로 변경
function makeDom(datas) {
    var dom = [];           // 새로 정렬될 데이터 배열
    var objects = {};       // replyId를 키로, reply를 값으로 하는 맵

    datas.forEach(data=>{
        data.regDate = new Date(data.regDate);
        data.children = [];           // 하위 댓글을 위한 배열
        objects[data.replyId] = data; // 맵 등록
        if(data.parent == 0) {        // 최상위 댓글
            dom.push(data);           // dom 배열에 추가
        } else {
            // 하위 댓글인 경우 부모의 children 배열에 추가
            objects[data.parent].children.push(data);
        }
    })

    return dom;
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
//트리 구조를 탐색하여 Media Object 구조 생성 - 재귀호출
function makeMediaObject(writer, item) {
  // 단말(leaf) reply (자식이 없는 reply)
  if(item.children.length==0) {
    return replyTempl.mediaObjectTempl(item, writer);
  }

  // 중간 reply
  var self = replyTempl.mediaObjectTempl(item, writer);

  // 자식 reply에 대해서 재귀 호출
  item.children.forEach(child=>{
    var child = makeMediaObject(writer, child);
    self.find('.children').append(child);
  });
  return self;
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
$.fn.reflyList = function(opt) {  
    var self = this;  
  
    // 초기 목록 구성  
    opt.api.list('', function(datas) {  
        // 레벨순, 등록역순 배열을 트리구조 배열로 변환  
        makeDom(datas).forEach(  
            item=>self.append(  
                makeMediaObject(opt.writer, item)  
            )  
        );  
    });  
});  
  
return this;  
}
```

replyList 플러그인

❖ 하위 댓글 추가

**admin**
abcd

작성일:2018-04-17

↶

하위 댓글 쓰기

**abc**
abc가 댓글을 답니다.

또 댓글을 답니다.

작성일:2018-04-17

↶

✎

🗑

하위 댓글 등록

↶

↷

작성 취소

**admin**
댓글 첫번째 테스트2

작성일:2018-04-17

↶

replyList 플러그인

❖ 댓글 관련 템플릿 객체: /resources/js/reply.templ.js

```
var replyTempl = {  
  replyAddTempl : ...,  
  mediaObjectTempl : ...,  
  buttonsTempl : ...,  
  // 하위 댓글 추가 템플릿  
  addTempl : `  
    <textarea></textarea>  
    <button class="reply-add">  
      <i class="fa fa-share-square"></i></button>  
    <button class="reply-cancel">  
      <i class="fa fa-undo"></i></button>`,  
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
$.fn.reflyList = function(opt) {  
    var self = this;  
    :  
    // 댓글 쓰기 패널 보여주기, this는 reply 버튼  
    function showReplyPanel() {  
        var parent = $(this).closest('.media-body');  
        parent.children('.work')  
            .append(replyTempl.addTempl);    // 하위 댓글 작성 추가  
    }  
  
    // 댓글 쓰기 패널 제거하기, this는 취소 버튼  
    function hideReplyPanel() {  
        $(this).parent().empty();  
    }  
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
// 댓글 등록, this는 하위댓글 등록 버튼
function createReply() {
    var obj = $(this).closest('.media');
    var reply = {
        boardId      : opt.boardId,
        writer       : opt.writer,
        parent       : parseInt(obj.data('reply-id')),
        replyLevel   : parseInt(obj.data('reply-level'))+1,
        content      : obj.find('textarea').val()
    }
    opt.api.create(reply,function(result){
        if(result) {
            result.regDate = new Date(result.regDate);
            obj.find('.children')
                .prepend(replyTempl.mediaObjectTempl(result,
                                                            reply.writer));
            obj.find('.work').empty();
        } else {
            alert('댓글 쓰기 실패')
        }
    });
}
```

replyList 플러그인


❖ 댓글 목록 플러그인: /resources/js/reply.js

```
// 하위 댓글 추가 관련
this.on('click', '.reply-add-show', showReplyPanel);
this.on('click', '.reply-cancel', hideReplyPanel);
this.on('click', '.reply-add', createReply);

return this;
}
```

replyList 플러그인


❖ 댓글 수정

**admin**
abcd

작성일:2018-04-17

↶

댓글 수정

**abc**

작성일:2018-04-17

↶

✎

🗑


abc가 댓글을 답니다.

↶

↷

작성 취소

댓글 수정 등록

**abc**

작성일:2018-04-17

↶

✎

🗑

또 댓글을 답니다.

replyList 플러그인

❖ 댓글 관련 템플릿 객체: /resources/js/reply.templ.js

```
var replyTempl = {  
  replyAddTempl : ...,  
  mediaObjectTempl : ...,  
  buttonsTempl : ...,  
  addTempl : ...,  
  editTempl : `  
    <textarea></textarea>  
    <button class="reply-edit">  
      <i class="fa fa-share-square"></i></button>  
    <button class="reply-edit-cancel">  
      <i class="fa fa-undo"></i></button>`,  
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
$.fn.reflyList = function(opt) {  
    var self = this;  
    :  
  
    // 댓글 수정 창 보이기, this는 수정 버튼  
    function showEditPanel() {  
        var content = $(this).closest('.media-body ')  
            .children('.reply-content');  
        var text = content.text();    // 현재 내용 추출  
        content.empty()                // 현재 내용 화면 제거  
            .data('old', text)    // 취소시 복원용 이전 데이터  
            .append(replyTempl.editTempl); // 수정창 추가  
        content.find('textarea').val(text);    // 수정창에 데이터 설정  
    }  
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
// 댓글 수정 창 제거, this는 취소 버튼
function hideEditPanel() {
    var text = $(this).parent()
        .data('old'); // 이전 데이터 추출
    $(this).parent()
        .empty()      // 현재 내용 제거
        .text(text);  // 이전 데이터로 복원
}
```


replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
// 댓글 수정 등록, this는 등록 버튼
function editReply() {
    var obj = $(this).parent();
    var reply = {
        replyId : $(this).closest('.media').data('reply-id'),
        content : obj.find('textarea').val()
    };

    api.update(reply, function(result){
        if(result) {
            result.regDate = new Date(result.regDate);
            obj.empty().text(reply.content);
        } else {
            alert('수정 실패');
        }
    });
}
```

replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
// 댓글 수정 관련 이벤트 핸들러 등록
this.on('click', '.reply-edit-show', showEditPanel);
this.on('click', '.reply-edit-cancel', hideEditPanel);
this.on('click', '.reply-edit', editReply);

return this;
}
```

replyList 플러그인

❖ 댓글 삭제



admin

abcd

작성일:2018-04-17



abc

삭제된 글입니다.

삭제된 글 표시

작성일:2018-04-17

삭제된 글은 버튼 없음



abc

또 댓글을 답니다.

작성일:2018-04-17



admin

댓글 첫번째 테스트2

작성일:2018-04-17



replyList 플러그인

❖ 댓글 목록 플러그인: /resources/js/reply.js

```
$.fn.reflyList = function(opt) {  
    var self = this;  
    :  
    //댓글 삭제 이벤트 핸들러, this는 삭제 버튼  
    function deleteContent() {  
        var self = $(this);  
        var media = $(this).closest('.media');  
        opt.api.remove(media.data('reply-id'), // replyId 추출  
            function(result){  
                if(result) { // 삭제 성공시  
                    media.find('.reply-content').eq(0)  
                        .text('삭제된 글입니다.');                    self.parent().remove(); // 버튼 그룹 제거  
                } else {  
                    alert('삭제 실패');                }  
            });  
    };  
    this.on('click', '.reply-delete',deleteContent);  
}
```