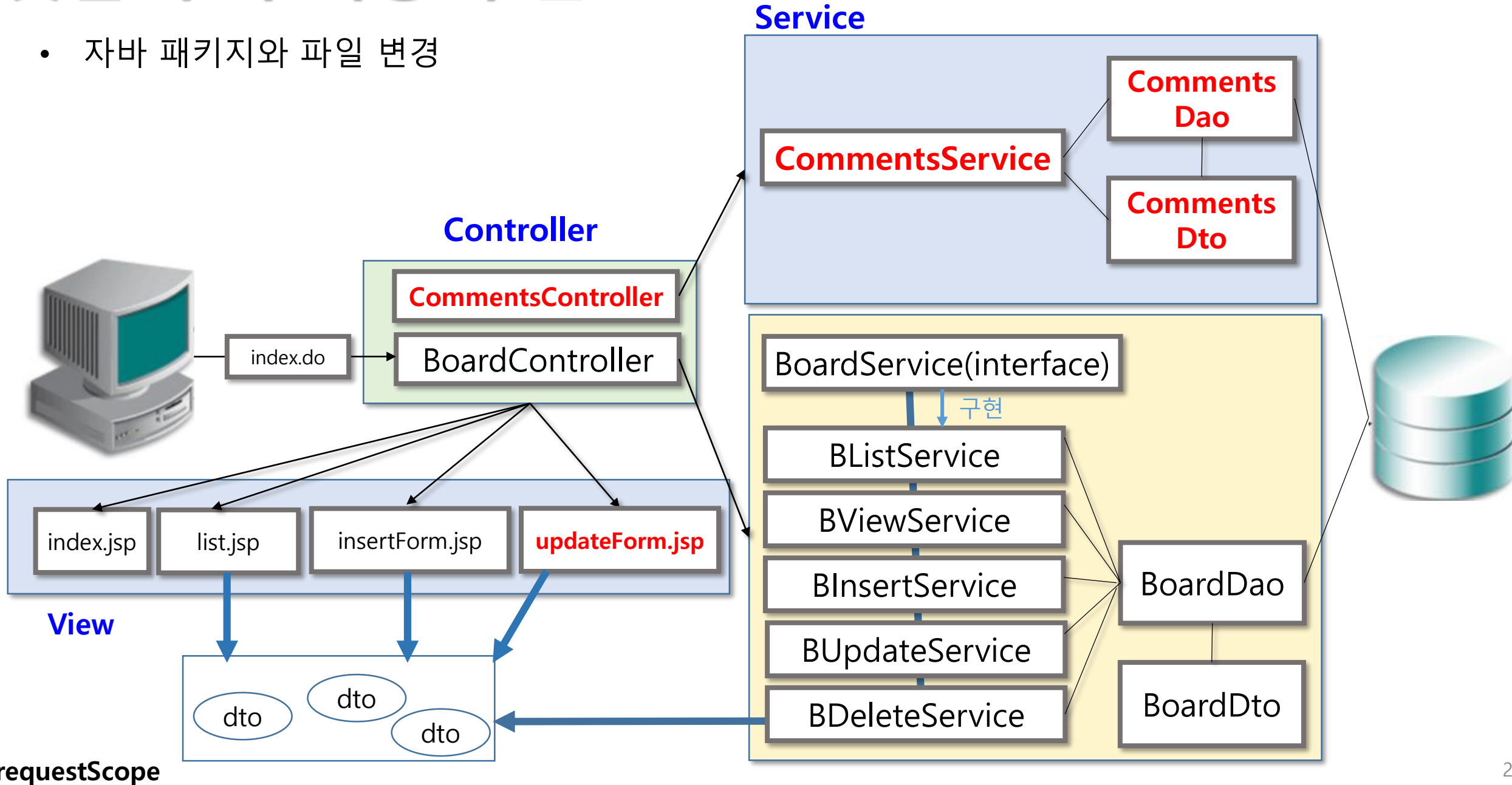


# Ajax로 한 줄 댓글 기능 구현

동의과학대학교 컴퓨터정보과  
김진숙

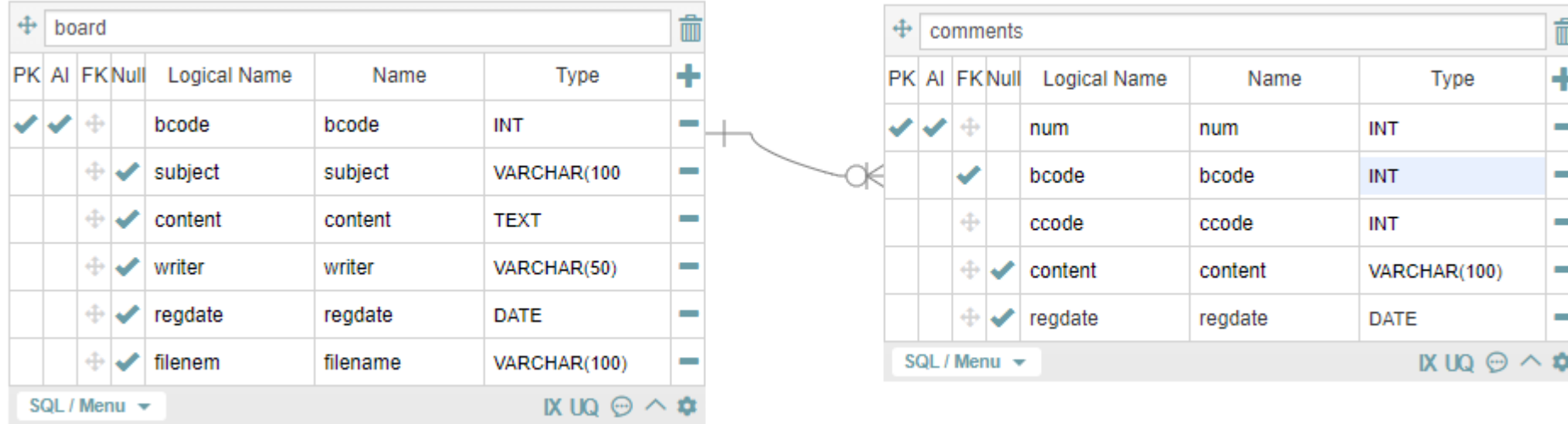
# 댓글 추가 기능 구현

- 자바 패키지와 파일 변경



# 준비 사항 - 데이터 테이블

- 댓글 테이블(comments)작성

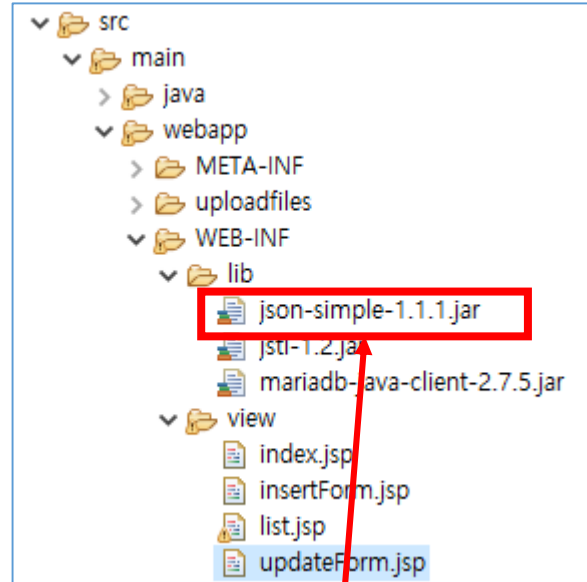


```
CREATE TABLE comments(  
    num INT AUTO_INCREMENT PRIMARY KEY,  
    BCODE INT,  
    CCODE INT,  
    CONTENT VARCHAR(100),  
    REGDATE DATE  
);
```

로그인/로그아웃 기능이 있는 경우  
WRITER 정보를 세션에 저장한 후 DB에 넣음

# 준비 사항 - JSON 라이브러리

- json-simple-1.1.1.jar



- <http://mvnrepository.com> 에서 json simple로 검색하여 라이브러리 다운로드
- WEB-INF/lib에 추가

# 실습 - 댓글 추가 기능 구현하기

| num | subject |
|-----|---------|
| 107 | Hi      |
| 94  | 반가워요^^  |
| 93  | 오늘은 불금  |
| 92  | 오늘은 불금  |
| 90  | 안녕49    |
| 89  | 안녕48    |
| 88  | 안녕47    |

### 상세보기

|          |           |         |            |
|----------|-----------|---------|------------|
| bcode    | 107       | subject | Hi         |
| content  | 잘 있었어요??? |         |            |
| writer   | 홍길동       | regDate | 2022-10-16 |
| filename | 진달래.gif   |         |            |

게시글 수정 | 게시글 삭제 | 게시글 목록 | 게시글 등록

댓글 :  한줄 댓글 작성

|   |             |            |
|---|-------------|------------|
| 2 | 지난 주에 봤죠??? | 2022-10-16 |
| 1 | 반가워요        | 2022-10-16 |

# CommentsDto.java

```
1 package cs.dit.board;
2
3 import java.sql.Date;
4
5 public class CommentsDto {
6     private int num;
7     private int ccode;
8     private int bcode;
9     private String content;
10    private Date regDate;
11
12    public int getNum() {
13        return num;
14    }
15    public void setNum(int num) {
16        this.num = num;
17    }
18    public int getCcode() {
19        return ccode;
20    }
21    public void setCcode(int ccode) {
22        this.ccode = ccode;
23    }
24    public int getBcode() {
25        return bcode;
26    }
```

```
27    public void setBcode(int bcode) {
28        this.bcode = bcode;
29    }
30    public String getContent() {
31        return content;
32    }
33    public void setContent(String content) {
34        this.content = content;
35    }
36    public Date getRegDate() {
37        return regDate;
38    }
39    public void setRegDate(Date regDate) {
40        this.regDate = regDate;
41    }
42    //생성자
43    public CommentsDto(int num, int ccode, int bcode, String content, Date regDate) {
44        this.num = num;
45        this.ccode = ccode;
46        this.bcode = bcode;
47        this.content = content;
48        this.regDate = regDate;
49    }
50    public CommentsDto() {}
51 }
```

# CommentsController.java

```
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.annotation.WebServlet;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 import org.json.simple.JSONArray; json-simple-1.1.1.jar 라이브러리
11
12 @WebServlet("*.ct") .ct 로 요청
13 public class CommentsController extends HttpServlet {
14     private static final long serialVersionUID = 1L;
15
16     protected void doGet(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18         request.setCharacterEncoding("utf-8");
19         response.setContentType("text/html; charset=utf-8");
20
21         String uri = request.getRequestURI(); //uri : /member-mvc-comments/list.do
22
23         String com= uri.substring(uri.lastIndexOf("/") + 1, uri.lastIndexOf(".ct")); //command : insert
24
25         CommentsService cService = new CommentsService();
26
27         if(com != null && com.trim().equals("cList")) {
28             System.out.println("cList");
29             cService.listComments(request, response);
30
31         } else if(com != null && com.trim().equals("cInsert")) {
32             System.out.println("cInsert");
33             cService.insertComments(request, response);
34         }
35     }
```

# CommentsService.java

```
1) import org.json.simple.JSONArray; json-simple-1.1.1.jar 라이브러리
11
12 public class CommentsService {
13
14     CommentsDao dao = new CommentsDao(); //아래의 메소드에서 공통으로 사용하는 객체
15
16     public void listComments(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18
19         int bcode = Integer.parseInt(request.getParameter("bcode"));
20         response.setContentType("application/x-json;charset=utf-8");
21
22         JSONArray list= new JSONArray();
23
24         list= dao.listComments(bcode);
25
26         response.getWriter().print(list); //ajax에게 전달
27     }
28
29     public void insertComments(HttpServletRequest request, HttpServletResponse response)
30         throws ServletException, IOException {
31
32         int bcode = Integer.parseInt(request.getParameter("bcode"));
33
34         String comments = request.getParameter("comments");
35
36         CommentsDto dto = new CommentsDto(0, 0, bcode, comments, null); //db에 입력할 dto객체 생성
37
38         dao.insertComments(dto);
39     }
40 }
```

← 리스트 출력

← 댓글 입력



# CommentsDao.java – getConnection()

```
1 package cs.dit.board;
2
3 import java.sql.CallableStatement;
4 import java.sql.Connection;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7
8 import javax.naming.Context;
9 import javax.naming.InitialContext;
10 import javax.sql.DataSource;
11
12 import org.json.simple.JSONArray;
13 import org.json.simple.JSONObject;
14
15 public class CommentsDao {
16     /**=====
17      * 패키지명 : cs.dit.board
18      * 파일명 : CommentsDao.java
19      * 작성자 : 김진숙
20      * 변경이력 :
21      * 2022-10-05/ 최초작성/ 김진숙
22      * 프로그램 설명 : comments 테이블과 내용과 연동하여 한줄답변 관리
23      * - getConnection() : 커넥션풀에서 연결객체 얻기
24      * - listComments(int bcode)
25      * - insertComments(CommentsDto dto)
26      *=====*/
27     private Connection getConnection() throws Exception{
28         Context initCtx = new InitialContext();
29         Context envCtx = (Context) initCtx.lookup("java:comp/env");
30         DataSource ds = (DataSource) envCtx.lookup("jdbc/jskim");
31         Connection con = ds.getConnection();
32
33         return con;
34     }
}
```

json-simple-1.1.1.jar 라이브러리

# CommentsDao.java – listComments()

```
36 //Comments 관련 코드
37 public JSONArray listComments(int bcode) {
38     String sql = "select ccode, content, regdate from comments where bcode=? order by ccode desc";
39     JSONArray list = new JSONArray();
40
41     try {
42         Connection con = getConnection();
43         PreparedStatement pstmt = con.prepareStatement(sql);
44         { pstmt.setInt(1, bcode);
45           ResultSet rs= pstmt.executeQuery();
46
47           while(rs.next()) {
48               // JSON으로 저장 : [{"regdate": "2022-10-01", "rcode": 1, "content": "저도 안녕"}]
49               JSONObject json = new JSONObject();
50               json.put("ccode", rs.getInt("ccode"));
51               json.put("content", rs.getString("content"));
52               String regdate = (rs.getDate("regdate")).toString();
53               json.put("regdate", regdate);
54
55               list.add(json);
56           }
57           rs.close();
58       } catch (Exception e) {
59           e.printStackTrace();
60       }
61
62       return list;
63     }
```

# CommentsDao.java – insertComments()

```
65 public void insertComments(CommentsDto dto) {  
66     String sql = "{call proc_comments(?, ?)}";  
67  
68     try (  
69         Connection con = getConnection();  
70         CallableStatement cstmt = con.prepareCall(sql);  
71     )  
72     {  
73         cstmt.setInt(1, dto.getBcode());    //SQL문과 데이터 바인딩  
74         cstmt.setString(2, dto.getContent());  
75  
76         cstmt.executeUpdate();  
77     } catch (Exception e) {  
78         e.printStackTrace();  
79     }  
80 }  
81 }
```

저장 프로시저 호출

# 한줄 댓글 저장을 위한 stored procedure

저장 프로시저 정의

-- 저장 프로시저 정의

DELIMITER \$\$

```
CREATE OR REPLACE PROCEDURE proc_comments(  
  IN _bcode INT,  
  IN _content VARCHAR(100)  
)  
BEGIN  
  DECLARE max_ccode INT;  
  
  SELECT MAX(ccode) into max_ccode FROM comments WHERE bcode = _bcode;  
  IF max_ccode IS NULL THEN  
    INSERT INTO comments(bcode, ccode, content, regdate) VALUES(_bcode, 1, _content, NOW());  
  ELSE  
    INSERT INTO comments(bcode, ccode, content, regdate) VALUES(_bcode, max_ccode+1, _content, NOW());  
  END IF;  
END $$
```

DELIMITER; -- 구분기호를 다시 세미콜론으로 변경하기 위해 사용 proc\_comments

-- 저장 프로시저 호출

CALL proc\_comments(107, '반갑습니다!!!');

- DELIMITER \$\$는 구문문자로 이 명령 없이 프로시저를 작성하면 세미콜론으로 끝나는 SQL 부분이 각각 나뉘어져 서버로 보내짐
- 프로시저를 작성할때는 항상 첫줄에는 DELIMITER \$\$를 선언하고 END에는 \$\$으로 프로시저 끝나는 부분을 표시하고 마지막에는 구분문자가 끝나는 DELIMITER ;를 선언해서 사용

# updateForm.jsp

- 댓글 출력 코드 추가

```
114< <div class="container">
115<   <table class="table" style="text-align:center; border: 1px solid #dddddd">
116<     <tr>
117<       <td style="background-color:#fafafa; text-align:center">댓글 : </td>
118<       <td><input class="form-control" type="text" id="comments" size="80"></td>
119<       <td colspan="2"><button class="btn btn-primary pull-right" onclick="cInsert();">한줄 댓글 작성</button></td>
120<     </tr>
121<   </table>
122<   <table class="table" style="text-align:center; border:1px solid #dddddd">
123<     <tbody id ="commentsTable">
124<     </tbody>
125<   </table>
126< </div>
127< </body>
128< </html>
```

# updateForm.jsp – javascript ajax

```
54 <!-- 댓글 달기 -->
55 <script type="text/javascript">
56     var xhr1 = new XMLHttpRequest(); //rlist
57     var xhr2 = new XMLHttpRequest(); //rinsert
58
59     var bcode = document.getElementById("bcode").value;
60
61     function cList(){
62         var commentsTable = document.getElementById("commentsTable");
63         commentsTable.innerHTML = "";
64
65         xhr1.onreadystatechange = function(){
66             if(this.readyState == 4 && this.status == 200){
67                 console.log(this.responseText);
68
69                 var json = this.responseText;
70                 var list = JSON.parse(json); // JSON 형식의 문자열을 자바스크립트 객체로 변환함.
71
72                 for(var i in list){
73                     var row = commentsTable.insertRow();
74                     var cell1 = row.insertCell(0);
75                     var cell2 = row.insertCell(1);
76                     var cell3 = row.insertCell(2);
77                     cell1.innerHTML = list[i].ccode;
78                     cell2.innerHTML = list[i].content;
79                     cell3.innerHTML = list[i].regdate;
80                 }
81             }
82         };
83         xhr1.open('POST', '/board-mvc-comments/cList.ct', true);
84         xhr1.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
85         xhr1.send('bcode=' + bcode);
86     }
```

← 한 줄 댓글 리스트 출력

# updateForm.jsp – javascript ajax

```
89  function cInsert(){
90      var comments = document.getElementById("comments").value;
91
92      xhr2.onreadystatechange = function(){
93          if(this.readyState == 4 && this.status == 200){
94              document.getElementById("comments").value="";
95              cList();
96          }
97      };
98
99      xhr2.open('POST', '/board-mvc-comments/cInsert.ct', true);
100     xhr2.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
101
102     var data = 'comments=' + comments + '&bcode='+ bcode;
103     xhr2.send(data);
104 }
105
106 window.onload = function(){ cList() };
107 </script>
```

한줄 댓글 입력

# updateForm.jsp – jquery ajax

```
<!-- 댓글 달기 -->
<script type="text/javascript">
var bcode = $('#bcode').val();

function cList(){
$.ajax({
url: './cList.ct?bcode=' + bcode,
type: 'post',
async: true,
dataType: 'json',
success: function(data){
var str = '';

for(var i in data){
str += '<tr><td>' + data[i].ccode + '<td>';
str += '<td>' + data[i].content + '<td>';
str += '<td>' + data[i].regdate + '<td><tr>';
}
$('#commentsTable').html(str);
}
});
}
```

← 한 줄 댓글 리스트 출력



# updateForm.jsp – jquery ajax

```
function cInsert(){  
    $.ajax({  
        url: './cInsert.ct?bcode=' + bcode,  
        type: 'post',  
        async: true,  
        data: {"comments": $('#comments').val()},  
        success: function (data){  
            $('#comments').val('');  
            cList();  
        }  
    });  
}  
  
window.onload = function(){cList();}
```



한줄 댓글 입력