



한림대학교 SW중심대학

MVC 모델

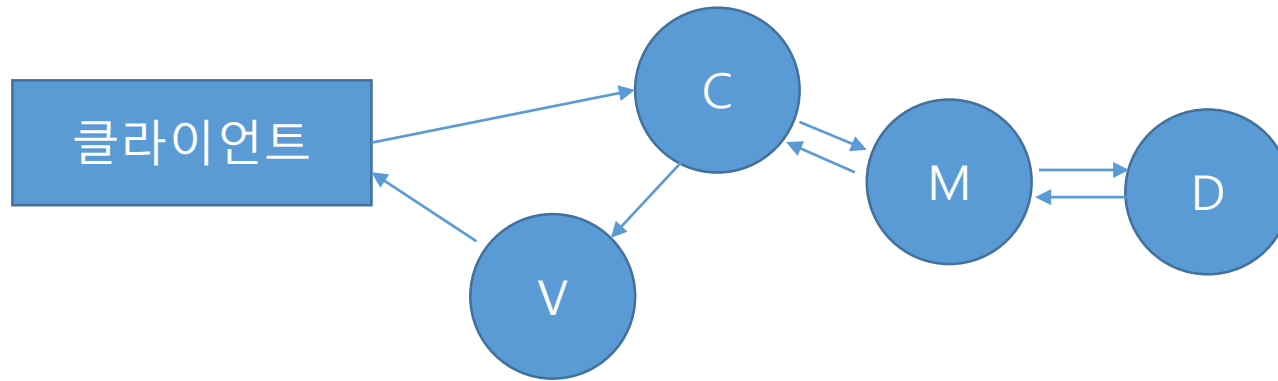
MVC란 Model, View, Controller를 뜻하는 용어로 개발 형태의 일종입니다.

Model은 데이터베이스와의 관계를 담당합니다. 클라이언트의 요청에서 필요한 자료를 데이터베이스로부터 추출하거나, 수정하여 Controller로 전달 합니다.

View는 사용자한테 보여지는 UI 화면 입니다. 주로 .jsp파일로 작성 하며, Controller에서 어떤 View 컴포넌트를 보여줄지 결정 합니다.

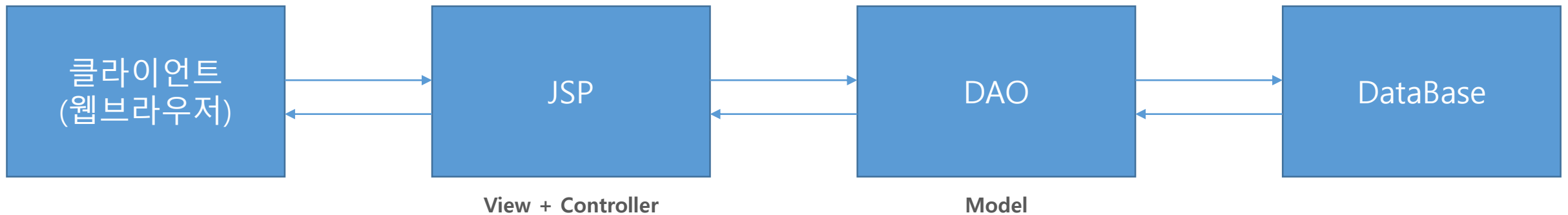
Controller는 클라이언트의 요청을 받고, 적절한 Model에 지시를 내리며, Model에서 전달된 데이터를 적절한 View에 전달 합니다.

이렇게 작업을 분할하면, 추후 유지보수에 좋습니다.

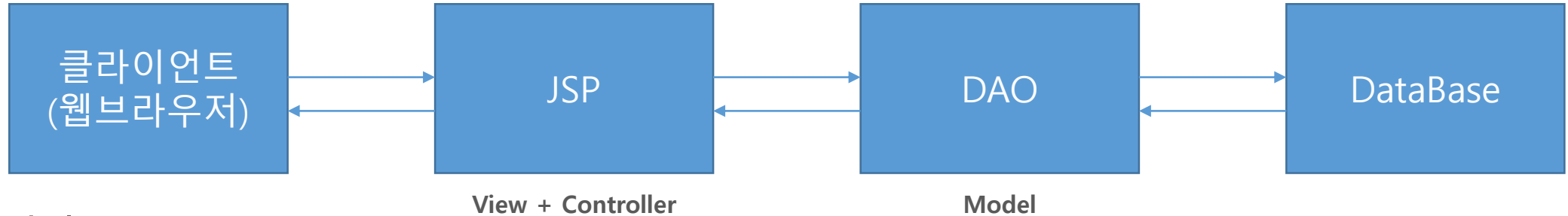


Model1

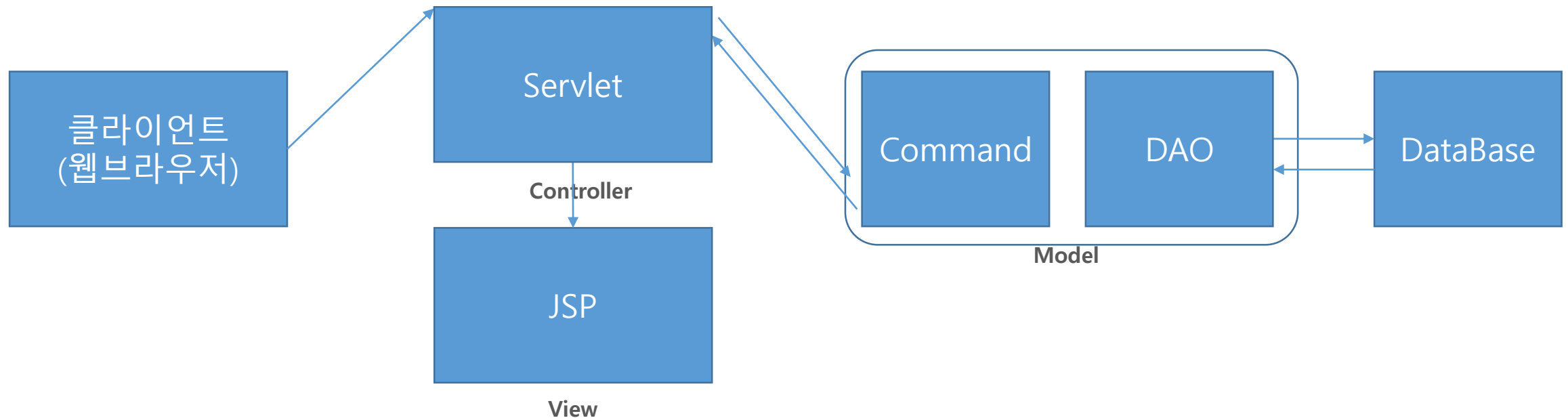
MVC에서 View와 Controller가 같이 있는 형태 입니다.



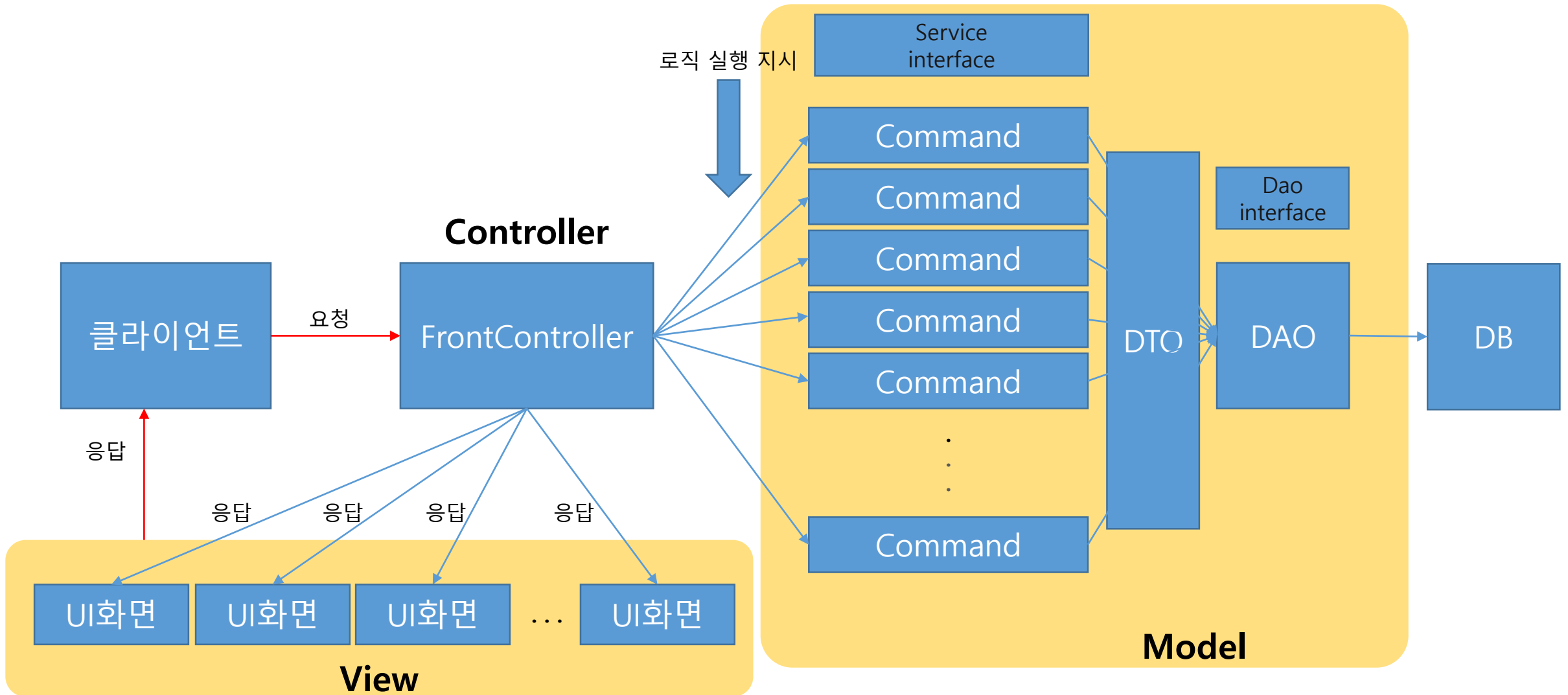
- Mode1
 - MVC에서 View와 Controller가 같이 있는 형태 입니다.



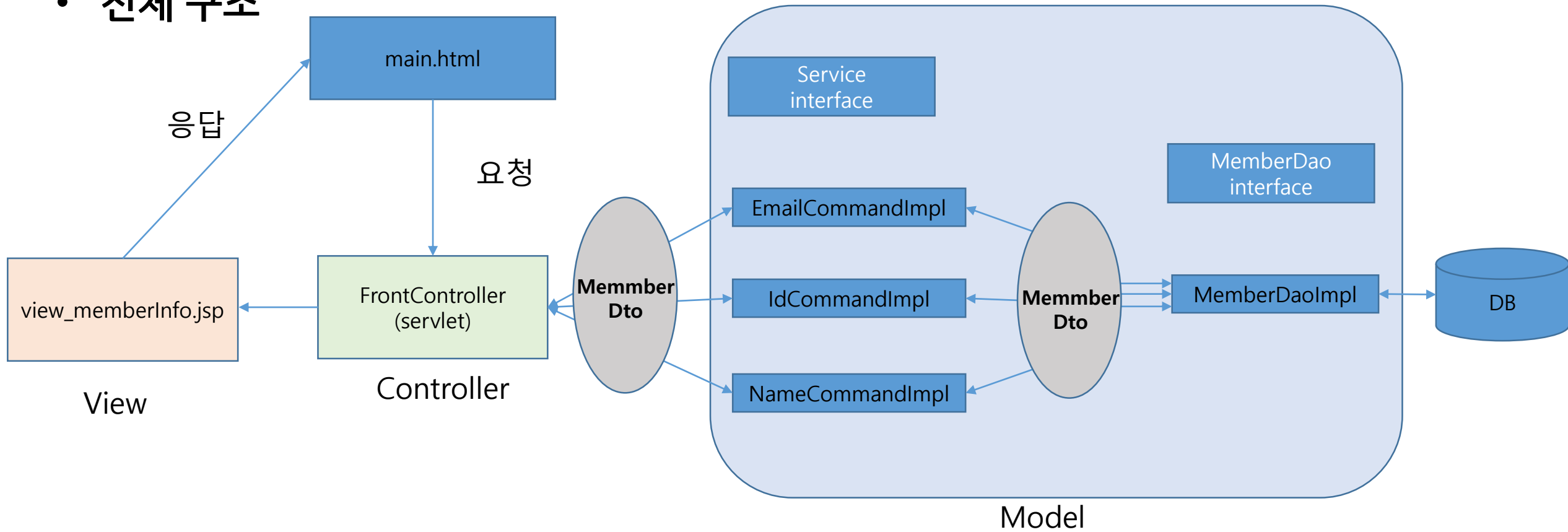
- Model2
 - MVC에서 Model, View 그리고 Controller가 모두 모듈화 되어 있는 형태 입니다.



전체적인 컴포넌트 설계



- 전체 구조



- 테이블 구조
 - table name : members

Column Name	Datatype	PK	NN
id	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
pw	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
name	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
reg_ate	TIMESTAMP(0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
address	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>

2021-04-08 11:31:41


```
CREATE TABLE `board` (  
  `bId` int(11) NOT NULL AUTO_INCREMENT,  
  `bName` varchar(45) DEFAULT NULL,  
  `bTitle` varchar(100) DEFAULT NULL,  
  `bContent` varchar(300) DEFAULT NULL,  
  `bDate` date DEFAULT NULL,  
  `bHit` int(11) DEFAULT NULL,  
  `bGroup` int(11) DEFAULT NULL,  
  `bStep` int(11) DEFAULT NULL,  
  `bIndent` int(11) DEFAULT NULL,  
  PRIMARY KEY (`bId`)  
)
```

bId	bName	bTitle	bContent	bDate	bHit	bGroup	bStep	bIndent
3	john	hi	my name is john	NULL	NULL	NULL	NULL	NULL
4	mary	hi john	see u tomorrow~	NULL	0	4	0	0


```
private void actionDo(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    System.out.println("actionDo");  
  
    request.setCharacterEncoding("EUC-KR");  
  
    String viewPage = null;  
    BCommand command = null;  
  
    String uri = request.getRequestURI();  
    String conPath = request.getContextPath();  
    String com = uri.substring(conPath.length());  
  
    if(com.equals("/write_view.do")) {  
        viewPage = "write_view.jsp";  
    }  
  
    if(com.equals("/write.do")) {  
        command = new BWriteCommand();  
        command.execute(request, response);  
        viewPage = "list.do";  
    }  
  
    RequestDispatcher dispatcher = request.getRequestDispatcher(viewPage);  
    dispatcher.forward(request, response);  
}
```

← 웹브라우저에서 글쓰기 UI화면

← 'write.do' 요청이 들어오면 해당 Command를 생성하여 적절한 로직을 실행 후 'list.do' 페이지로 포워딩

Command 만들기

Command 인터페이스를 이용해서 Command 클래스들을 만듭니다.

패키지 : com.javalec.ex.command

Command 인터페이스 : BCommand

Write Command 클래스 : BWriteCommand

```
BCommand.java ✖
1 package com.javalec.ex.command;
2
3 import javax.servlet.http.HttpServletRequest;
4
5
6 public interface BCommand {
7
8     void execute(HttpServletRequest request, HttpServletResponse response);
9
10 }
```

@Override

```
public void execute(HttpServletRequest request, HttpServletResponse response) {
    // TODO Auto-generated method stub

    String bTitle = request.getParameter("bTitle");
    String bName = request.getParameter("bName");
    String bContent = request.getParameter("bContent");

    BDao dao = new BDao();
    dao.write(bTitle, bName, bContent);
}
```

데이터 베이스와 연결하여 사용자가
입력한 내용을 DB에 Insert 합니다.

데이터 베이스의 데이터 DTO객체를 만듭니다.

패키지 : com.javalec.ex.dto

DTO 클래스 : BDto

```
public BDto(int bld, String bName, String bTitle, String bContent, Timestamp bDate, int bHit, int bGroup, int bStep, int blndent) {  
    // TODO Auto-generated constructor stub  
    this.bld = bld;  
    this.bName = bName;  
    this.bTitle = bTitle;  
    this.bContent = bContent;  
    this.bDate = bDate;  
    this.bHit = bHit;  
    this.bGroup = bGroup;  
    this.bStep = bStep;  
    this.blndent = blndent;  
}
```

← 생성자 및 속성 설정

```
public int getbld() {  
    return bld;  
}
```

← Getter, Setter

```
public void setbld(int bld) {  
    this.bld = bld;  
}
```

데이터 베이스에 연결하여 필요한 로직을 수행하는 DAO클래스를 만듭니다.

패키지 com.javalec.ex.dao

DTO 클래스 : BDao

```
public BDao() {  
    try {  
        Context context = new InitialContext();  
        Context envContext = (Context)context.lookup("java:comp/env");  
        dataSource = (DataSource)envContext.lookup("jdbc/MySQLDB");  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

← 생성자에서 DBCP를 만듭니다.

```
public void write(String bName, String bTitle, String bContent) {  
    // TODO Auto-generated method stub  
  
    Connection connection = null;  
    PreparedStatement preparedStatement = null;  
  
    try {  
        connection = dataSource.getConnection();  
        String query = "insert into mvc_board (bld, bName, bTitle, bCc  
        preparedStatement = connection.prepareStatement(query);  
        preparedStatement.setString(1, bName);  
        preparedStatement.setString(2, bTitle);  
        preparedStatement.setString(3, bContent);  
        int rn = preparedStatement.executeUpdate();
```

← DBCP로부터 Connection을 얻고, 데이터 베이스와 관련한 필요한 작업을 시작 합니다.

A Web Page

← → × 🏠 http:// 🔍

번호	이름	제목	날짜	히트
4	ho			
3	jonh			
3	apple	Patata		
1	Val	☑		

< 1 2 3 4 5 6 7 8 9 10 >

제목 ▼
제목+내용
이름
날짜

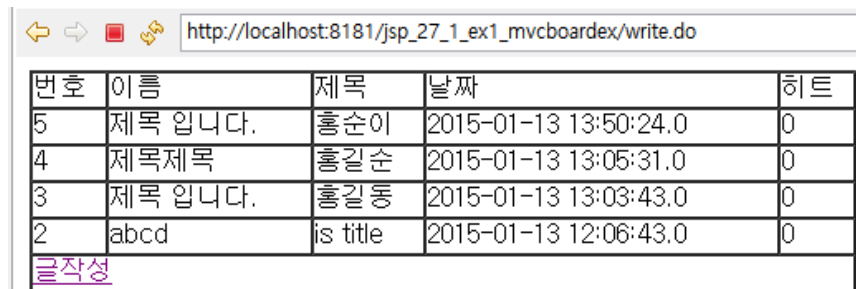
search

검색

클라이언트의 요청에 대해서 FrontController에서 작업을 분기하고, 해당 Command클래스가 작동하여 DAO를 이용한 데이터 베이스 작업을 합니다. DAO클래스의 결과물로 DTO객체가 View(jsp페이지)로 전달되며, View에서는 클라이언트의 요청에 대한 응답으로 화면(UI)를 구성하여 출력 합니다.

View 파일 : list.jsp

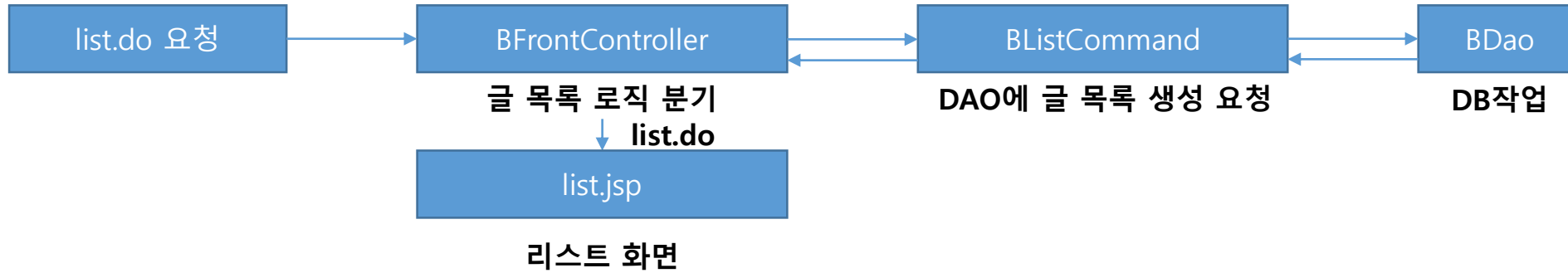
```
<table width="500" cellpadding="0" cellspacing="0" border="1">
  <tr>
    <td>번호</td>
    <td>이름</td>
    <td>제목</td>
    <td>날짜</td>
    <td>히트</td>
  </tr>
  <c:forEach items="${list}" var="dto">
    <tr>
      <td>${dto.bId}</td>
      <td>${dto.bName}</td>
      <td>${dto.bTitle}</td>
      <td>${dto.bDate}</td>
      <td>${dto.bHit}</td>
    </tr>
  </c:forEach>
</table>
```



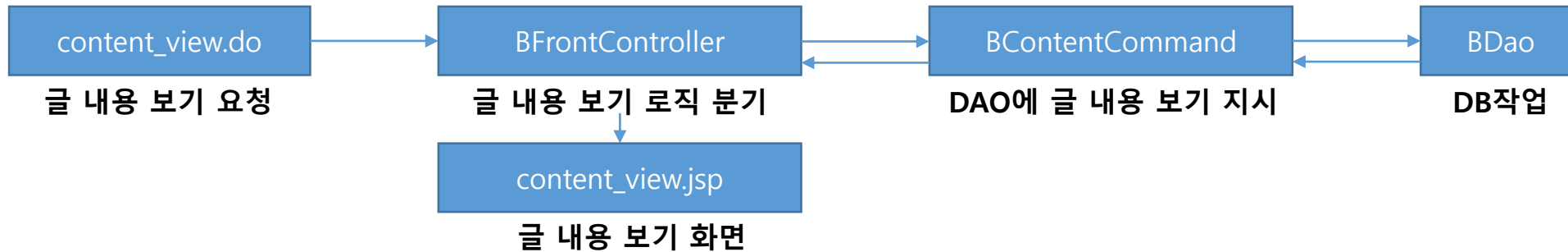
번호	이름	제목	날짜	히트
5	제목 입니다.	홍순이	2015-01-13 13:50:24.0	0
4	제목제목	홍길순	2015-01-13 13:05:31.0	0
3	제목 입니다.	홍길동	2015-01-13 13:03:43.0	0
2	abcd	is title	2015-01-13 12:06:43.0	0

[글작성](#)

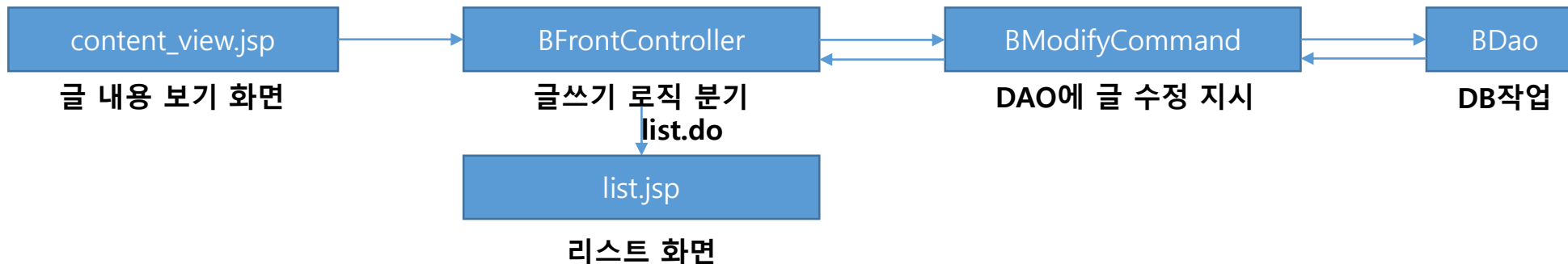
글 목록 페이지 만들기

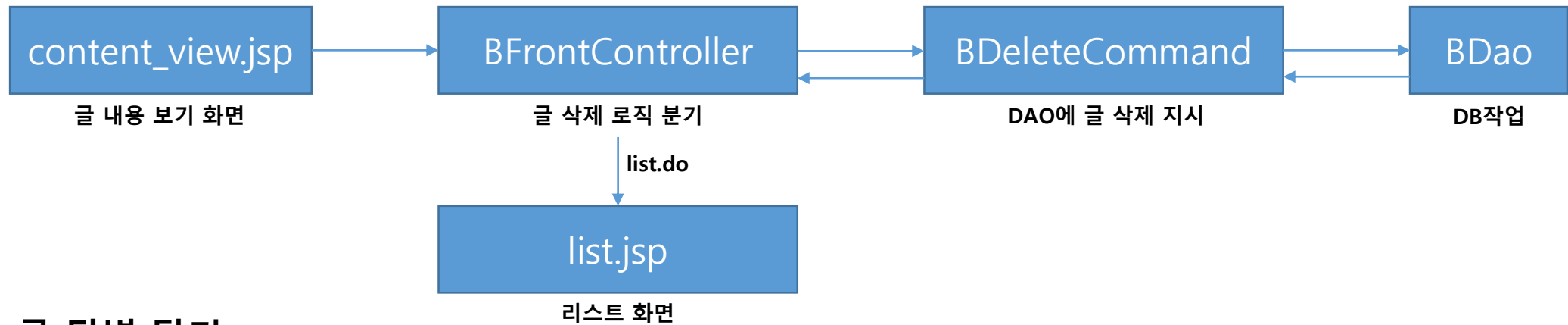


글 내용 보기 페이지 만들기

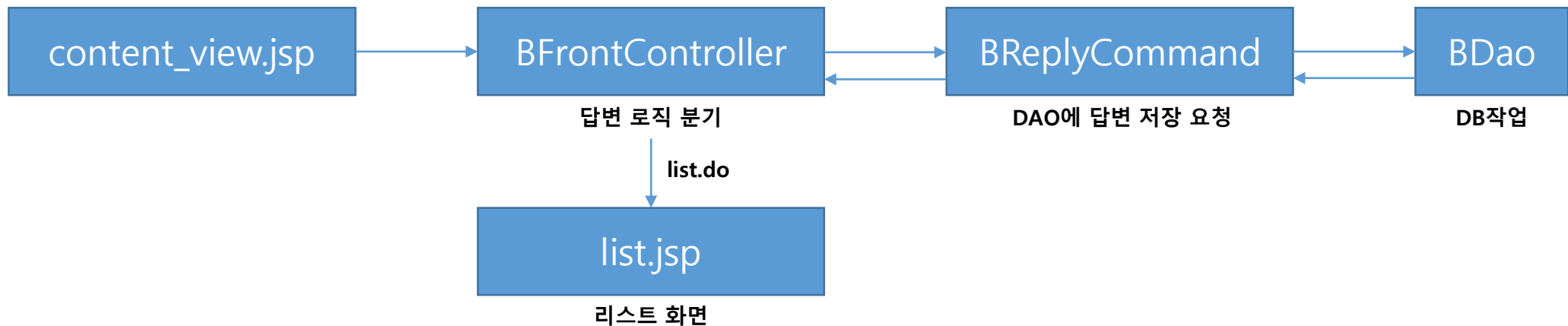


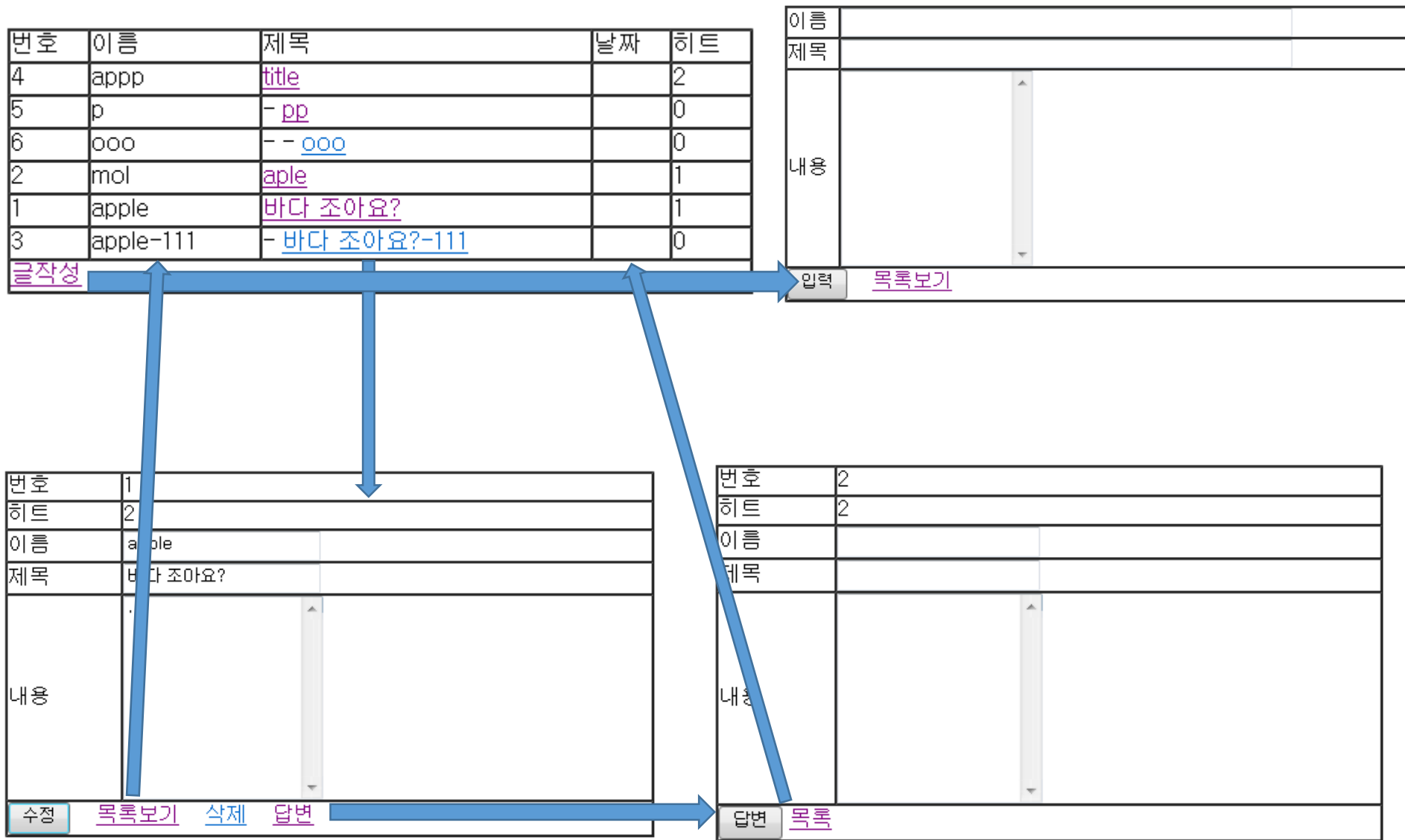
글 내용 수정 페이지 만들기





글 답변 달기





1. id, pw로 로그인후 게시글 작성 가능
2. 글쓴 사람만 수정 가능하도록 변경
3. 세션을 이용해서 로그인했을 경우에는 바로 글 수정 가능하게
4. 로그인 안했을 경우에는 로그인을 하고 이후 글수정시 세션을 만들어서 로그인 없이 글 수정 가능하도록

- ALTER TABLE '테이블명' AUTO_INCREMENT='시작할값'