

Model2를 이용한 게시판

안 화 수

게시판

❖ 주요 기능

1. Connection Pool
2. 댓글 기능
3. request 객체 공유 설정
4. **Controller 클래스** : Java Servlet
5. **Model** = Service + DAO
Service, DTO, DAO 클래스
6. **View (화면 인터페이스)** : EL , JSTL 사용

게시판

❖ 게시판 테이블 생성

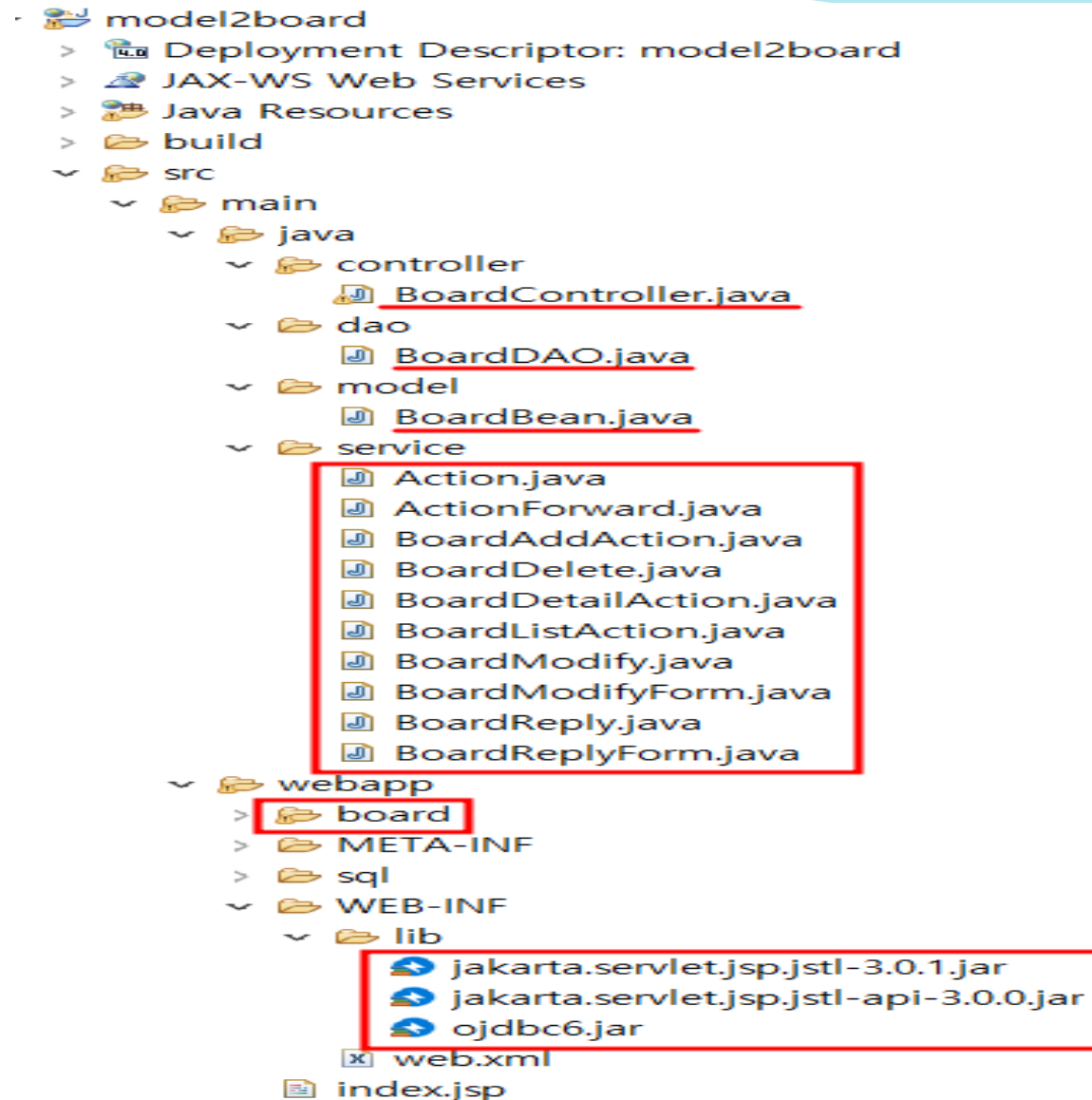
```
create table model2board(  
    board_num number primary key,  
    board_name varchar2(20),  
    board_pass varchar2(15),  
    board_subject varchar2(50),  
    board_content varchar2(2000),  
    board_re_ref number,  
    board_re_lev number,  
    board_re_seq number,  
    board_readcount number,  
    board_date timestamp );
```

```
create sequence model2board_seq  
start with 1  
increment by 1  
nocache;
```

```
-- 글 그룹번호  
-- 댓글 깊이 : 원문(0), 1, 2...  
-- 댓글 출력 순서 : 원문(0) 오름차순 정렬
```

게시판

❖ 프로젝트 구조



게시판

❖ 프로그램 주요 파일

model2board - src - main - java - controller - BoardController.java (Controller)

model - BoardBean.java (DTO)

dao - BoardDAO.java (DAO)

service -
Action.java (부모 인터페이스)
ActionForward.java (포워딩 설정)
BoardAddAction.java (글작성)
BoardListAction.java (글목록)
BoardDetailAction.java (상세 페이지)
BoardReplyForm.java (댓글 폼)
BoardReply.java (댓글)
BoardModifyForm.java (글수정 폼)
BoardModify.java (글수정)
BoardDeleteForm.java (글삭제 폼)
BoardDelete.java (글삭제)

게시판

❖ 프로그램 주요 파일

webapp - board -
board_write.jsp (글작성 폼)
board_list.jsp (글목록)
board_view.jsp (상세 페이지)
board_reply.jsp (댓글)
board_modify.jsp (글수정)
board_delete.jsp (글삭제)

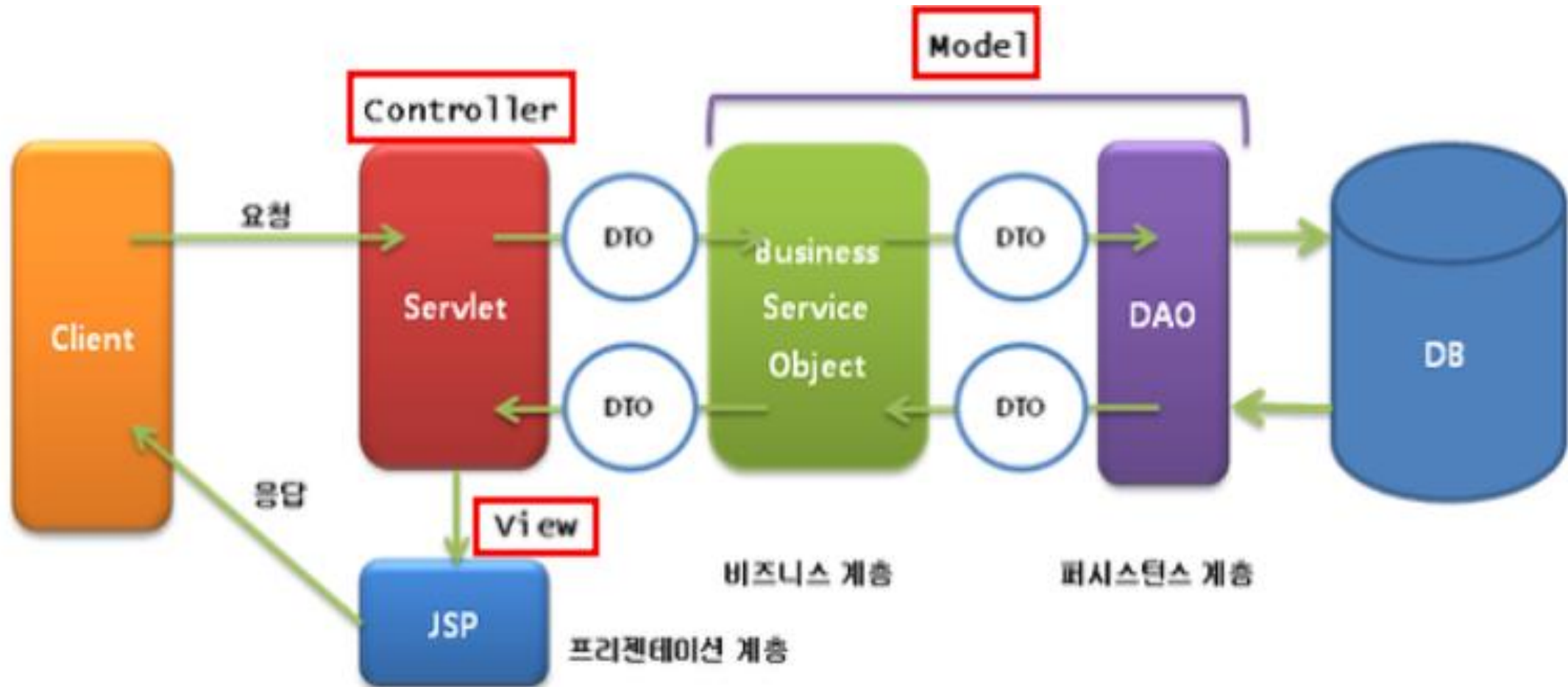
META-INF - context.xml (컨넥션풀 환경설정 파일)

WEB-INF - lib -
jakarta.servlet.jsp.jstl-3.0.1.jar
jakarta.servlet.jsp.jstl-api-3.0.0.jar
ojdbc6.jar

- web.xml (프로젝트의 환경설정파일)

모델2 설계

❖ MVC Architecture



controller 클래스

```
package controller;
```

```
import java.io.IOException;
```

```
@WebServlet("*.do") // do 확장자로 요청하는 모든 요청을 받는다는 의미
```

```
public class BoardController extends HttpServlet {
```

```
    // doGet(), doPost() 메소드의 공통적인 작업을 처리하는 메소드
```

```
    protected void doProcess(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
        String requestURI = request.getRequestURI();
```

```
        String contextPath = request.getContextPath();
```

```
        String command = requestURI.substring(contextPath.length());
```

```
        System.out.println("requestURI:" + requestURI); // /model2board/BoardAddAction.do
```

```
        System.out.println("contextPath:" + contextPath); // /model2board
```

```
        System.out.println("command:" + command); // /BoardAddAction.do
```

```
        Action action = null;
```

```
        ActionForward forward = null;
```

```
        // 글작성(원문작성)
```

```
        if(command.equals("/BoardAddAction.do")) {
```

```
            try {
```

```
                action = new BoardAddAction();
```

```
                forward = action.execute(request, response);
```

```
            } catch (Exception e) {
```

```
                e.printStackTrace();
```

```
            }
```


controller 클래스

```
// 포워딩 처리
if(forward != null) {
    if(forward.isRedirect()) { // redirect 방식으로 포워딩
        response.sendRedirect(forward.getPath());
    }else { // dispatcher 방식으로 포워딩
        RequestDispatcher dispatcher = request.getRequestDispatcher(forward.getPath());
        dispatcher.forward(request, response);
    }
}

} // doProcess() end

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    System.out.println("get");

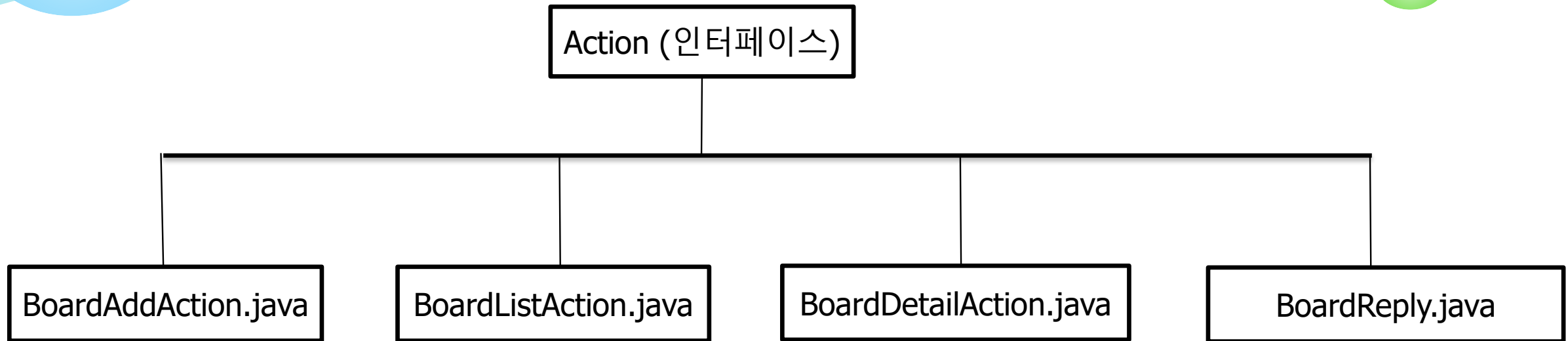
    doProcess(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    System.out.println("post");

    doProcess(request, response);
}

}
```

Service 클래스



- 구현 클래스들은 Action 인터페이스를 implements로 상속을 받고, 추상 메소드(execute())를 메소드 오버라이딩 해서 통일성 있게 구현한다.

Service 클래스

❖ Action 인터페이스

```
package service;
```

```
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;
```

```
public interface Action {
```

```
// 추상 메소드
```

```
public ActionForward execute(HttpServletRequest request, HttpServletResponse response)  
    throws Exception;
```

```
}
```

Service 클래스

❖ ActionForward 클래스
package service;

```
public class ActionForward {
```

```
    private boolean redirect;  
    private String path;
```

```
// 포워딩 방식 설정  
// 포워딩 페이지명 설정
```

```
    public boolean isRedirect() {  
        return redirect;  
    }
```

```
    public void setRedirect(boolean redirect) {  
        this.redirect = redirect;  
    }
```

```
    public String getPath() {  
        return path;  
    }
```

```
    public void setPath(String path) {  
        this.path = path;  
    }
```

```
}
```

Service 클래스

```
package service;

import dao.BoardDAO;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import model.BoardBean;

public class BoardAddAction implements Action {

    @Override
    public ActionForward execute(HttpServletRequest request, HttpServletResponse response) throws Exception {
        // TODO Auto-generated method stub
        System.out.println("BoardAddAction");

        request.setCharacterEncoding("utf-8");

        BoardBean board = new BoardBean();
        board.setBoard_name(request.getParameter("board_name"));
        board.setBoard_pass(request.getParameter("board_pass"));
        board.setBoard_subject(request.getParameter("board_subject"));
        board.setBoard_content(request.getParameter("board_content"));
```

Service 클래스

```
BoardDAO dao = BoardDAO.getInstance();  
int result = dao.insert(board); // 원문 글작성  
if(result == 1) System.out.println("글작성 성공");  
  
ActionForward forward = new ActionForward();  
forward.setRedirect(true);  
forward.setPath("./BoardListAction.do");  
  
return forward;  
}  
  
}
```