MVC2

1. Index.jsp

-response.sendRedirect("BoardServlet?command=board\_list");

Command에 board\_list 를 저장하고 BoardServlet로 이동한다

1. BoardList.jsp

-<link rel="stylesheet" type="text/css" href="css/shopping.css">

css폴더에 있는 shopping.css 스타일을 가져와 적용한다.

-href="BoardServlet?command=board\_write\_form">게시글 등록

Command에 board\_write\_form를 저장하고 BoardServlet로 이동한다

-<c:forEach var="board" items="${boardList }"></c:forEach>

Board에 "${boardList }"를 저장한다.

- <a href="BoardServlet?command=board\_view&num=${board.num}">${board.title } </a>

게시리스트의 제목을 클릭하면 Command에 board\_view&  
 num에${board.num}를 저장하고 BoardServlet로 이동한다.

1. boardWrite.jsp

<script type="text/javascript" src="script/board.js"></script>  
<input type="submit" value="등록"onclick="return boardCheck()">

script폴더에 있는 board.js를 불러와 적용시킨다. Board.js에 있는 boardCheck()를 불러와  
true or false를 리턴받아 true가 리턴되면 action으로 submit한다.

1. Boardupdate.jsp/boardView.jsp
2. BoardServlet

String command = request.getParameter("command");  
System.out.println("BoardServlet= : " + command);  
ActionFactory af=ActionFactory.getInstance();  
Action action=af.getAction(command);   
if(action != null){action.execute(request, response);  
 jsp페이지에서 받아온 “command”를 저장하고 ActionFactory의 객체 af를 만든다.  
af.getAction(command) 함수로 command에 따른 new 자식action클래스를 만들어 부모클래스 action으로 객체를 생성하고 execute함수로 페이지 이동을 한다.

1. ActionFactory

getAction(String command)함수  
if~else문으로 command의 값에 따른 Action클래스의 자식클래스를 생성한다.

1. 각 Action클래스

Execute(request,response)함수  
각 클래스별 기능을 하고 이동할 페이지의 URL을 받아  
RequestDispatcher dispatcher = request.getRequestDispatcher(url);  
dispatcher.forward(request, response); 디스패쳐로 각 URL에 따른 페이지로 이동한다.

1. DBManager

자바와 DB를 연결시키기 위한 클래스. 커넥션풀을 이용한다.

1. BoardVO

자바빈 클래스로서 자바와 DB를 연동할 때 데이터의 값을 저장하여 간단하게 데이터를 이용할 수 있게 한다.

1. BoardDAO

자바와 DB사이에서 명령과 결과를 주고받기 위한 클래스. 미리 입력해놓은 SQL문과 바인드 변수를 이용하여 함수를 호출하는 것만으로 쉽게 DB를 조작할 수 있다.