**[ 18 ] Front Controller와 Command 구조**

1. url-pattern
   1. 디렉터리 패턴 ; 디렉터리 형태로 서버의 해당 컴포넌트(서블릿)를 찾아서 실행하는 구조. 서블릿을 만들고 서블릿의 긴 주소가 보안상 문제가 되거나 너무 길어, 매핑을 한다. xml이나 어노테이션을 이용해 매핑을 할 때 사용했던 방식
      1. <http://localhost:8181/ch17/Hello> → /Hello 서블릿

http://localhost:8181/ch17/servlet/com.tj.ex.Hello 서블릿의 긴 주소를 매핑한다.

* + 1. <http://localhost:8181/ch17/Test> → /Test 서블릿
  1. 확장자 패턴 ; 확장자 형태로 서버의 해당 컴포넌트를 찾아서 실행하는 구조
     1. <http://localhost:8181/ch17/Hello>.do ↘
     2. <http://localhost:8181/ch17/Test>.do → \*.do 서블릿

1. FrontController ; 클라이언트의 다양한 요청을 한곳으로 집중시켜, 개발 및 유지보수에 효율성을 극대화

위의 경우 컴포넌트들끼리 중복되는 코드가 있을 수도 있고, 파일이 여러 개라 집중성이 떨어질 수 있다.

요청을 받는 창구를 하나로 집중시켰다고도 할 수 있다.

요청1

요청2

요청3

요청1 처리 컴포넌트

요청2 처리 컴포넌트

요청3 처리 컴포넌트

DAO

요청1

요청2

요청3

모든 요청을

처리하는 서블릿

DAO

예제 Ex0.jsp

<body>

<%

String uri = request.getRequestURI();

String conPath = request.getContextPath();

String command = uri.substring(conPath.length());

%>

URI : <%=uri %><br>

conPath : <%=conPath %>

command : <%=command %>

</body>

예제 ex1\_command.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"* %>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<h1><a href=*"insert.do"*>INSERT</a></h1>

<h1><a href=*"http://localhost:8181/ch18/select.do"*>SELECT</a></h1>

<h1><a href=*"/ch18/update.do"*>UPDATE</a></h1>

<h1><a href=*"*${conPath }*/delete.do"*>DELETE</a></h1>

</body>

</html>

FrontController.java

@WebServlet("\*.do")

**public** **class** FrontController **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

actionDo(request, response);

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

request.setCharacterEncoding("utf-8");

actionDo(request, response);

}

**private** **void** actionDo(HttpServletRequest request, HttpServletResponse response) **throws** IOException {

String uri = request.getRequestURI(); // "/ch18/insert.do"

String conPath = request.getContextPath(); // "/ch18"

String command = uri.substring(conPath.length()); // "/insert.do"

**if**(command.equals("/insert.do")) {

System.***out***.println("insert command 호출");

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = "+uri+"<br>");

writer.println("conPath = "+conPath+"<br>");

writer.println("들어온 요청은 "+command+"<br>");

writer.println("</body></html>");

writer.close();

}**else** **if**(command.equals("/select.do")) {

System.***out***.println("select command 호출");

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = "+uri+"<br>");

writer.println("conPath = "+conPath+"<br>");

writer.println("들어온 요청은 "+command+"<br>");

writer.println("</body></html>");

writer.close();

}**else** **if**(command.equals("/update.do")) {

System.***out***.println("update command 호출");

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = "+uri+"<br>");

writer.println("conPath = "+conPath+"<br>");

writer.println("들어온 요청은 "+command+"<br>");

writer.println("</body></html>");

writer.close();

}**else** **if**(command.equals("/delete.do")) {

System.***out***.println("delete command 호출");

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = "+uri+"<br>");

writer.println("conPath = "+conPath+"<br>");

writer.println("들어온 요청은 "+command+"<br>");

writer.println("</body></html>");

writer.close();

}

}

}

1. Command 구조 ; 클라이언트로부터 받은 요청들에 대해서, 서블릿이 작업을 직접 처리 하지 않고, 해당 클래스가 처리하도록 한다. FrontController 한군데서 처리하면 소스가 또 너무 방대해진다. FrontController.javj는 큰 형태만 갖고 있고, 세부적인 처리는 command에서 처리한다.

요청1

요청2

요청3

모든 요청을

직접 처리하는 서블릿

DAO

요청1

요청2

요청3

모든 요청을

받아 직접 처리하지 않고 command측에서 처리하도록 함

DAO

요청1 해당 처리 class

요청2 해당 처리 class

요청3 해당 처리 class

**Interface :** Service.java

사용법통일화

membersAll.do FrontController.java MembersAllService.java

memberJoin.do

직접 처리하지 않고 command클래스를 하나씩 만들어 준다

예제

**index.jsp**

<body>

<h1><a href=*"membersAll.do"*>멤버리스트 출력</a></h1>

<h1><a href=*"join.jsp"*>회원가입</a></h1>

</body>

**MemberDto.java**

**package** com.tj.ex;

**import** java.sql.Date;

**import** java.sql.Timestamp;

**public** **class** MemberDto {

**private** String id;

**private** String pw;

**private** String name;

**private** String email;

**private** Timestamp birth;

**private** Date rdate;

**private** String address;

**public** MemberDto() { }

**public** MemberDto(String id, String pw, String name, String email, Date birth, Date rdate, String address) {

**this**.id = id;

**this**.pw = pw;

**this**.name = name;

**this**.email = email;

**this**.birth = birth;

**this**.rdate = rdate;

**this**.address = address;

}

**Getters & setters**

@Override

**public** String toString() {

**return** "MemberDto [id=" + id + ", pw=" + pw + ", name=" + name + ", email=" + email + ", birth=" + birth

+ ", rdate=" + rdate + ", address=" + address + "]";

}

}

**MemberDao.java**

**package** com.tj.ex;

//getMembersAll / joinMember

**import** java.sql.Connection;

**import** java.sql.Date;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Timestamp;

**import** java.util.ArrayList;

**import** javax.naming.Context;

**import** javax.naming.InitialContext;

**import** javax.naming.NamingException;

**import** javax.sql.DataSource;

**public** **class** MemberDao {

**public** **static** **final** **int** ***FAIL*** = 0; // 실패 (회원가입, 정보수정)

**public** **static** **final** **int** ***SUCCESS*** = 1; // 성공(회원가입, 정보수정)

**private** **static** MemberDao *instance* = **new** MemberDao();;

**public** **static** MemberDao getInstance() {

**return** *instance*;

}

**private** MemberDao() {}

// getConnection() : conn객체 얻어오기

**private** Connection getConnection() **throws** SQLException {

Connection conn=**null**;

**try** {

Context ctx = **new** InitialContext();

DataSource ds =

(DataSource) ctx.lookup("java:comp/env/jdbc/Oracle11g");

conn = ds.getConnection();

} **catch** (NamingException e) {

System.***out***.println(e.getMessage());

}

**return** conn;

}

//1. 가입한 모든 회원 정보 가져오기

**public** ArrayList<MemberDto> getMembersAll(){

ArrayList<MemberDto> members = **new** ArrayList<MemberDto>();

Connection conn = **null**;

PreparedStatement pstmt = **null**;

ResultSet rs = **null**;

String sql = "SELECT \* FROM MEMBER";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

rs = pstmt.executeQuery();

**while**(rs.next()){

String id = rs.getString("id");

String pw = rs.getString("pw");

String name = rs.getString("name");

String email = rs.getString("email");

Timestamp birth = rs.getTimestamp("birth");

Date rDate = rs.getDate("rDate");

String address = rs.getString("address");

members.add(**new** MemberDto(id, pw, name, email, birth, rDate, address));

}

}**catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(rs!=**null**) rs.close();

**if**(pstmt!=**null**) pstmt.close();

**if**(conn!=**null**) conn.close();

} **catch** (Exception e) {}

}

**return** members;

}

// 2. 회원가입

**public** **int** insertMember(MemberDto dto) {

**int** result = ***FAIL***;

Connection conn = **null**;

PreparedStatement pstmt = **null**;

String sql = "INSERT INTO MEMBER "

+ "(ID, PW, NAME, EMAIL, BIRTH, RDATE, ADDRESS) "

+ "VALUES (?,?,?,?,?, SYSDATE, ?)";

**try** {

conn = getConnection();

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, dto.getId());

pstmt.setString(2, dto.getPw());

pstmt.setString(3, dto.getName());

pstmt.setString(4, dto.getEmail());

pstmt.setTimestamp(5, dto.getBirth());

pstmt.setString(6, dto.getAddress());

result = pstmt.executeUpdate();

System.***out***.println(result==***SUCCESS***? "회원가입성공":"회원가입실패");

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(pstmt!=**null**) pstmt.close();

**if**(conn !=**null**) conn.close();

} **catch** (Exception e) {System.***out***.println(e.getMessage());}

}

**return** result;

}

}

**Service.java**

**public** **interface** Service {

**public** ArrayList<MemberDto> execute(

HttpServletRequest request, HttpServletResponse response);

}

**MemberJoinService.java**

**package** com.tj.ex;

**import** java.sql.Timestamp;

**import** java.util.ArrayList;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** MemberJoinService **implements** Service {

@Override

**public** ArrayList<MemberDto> execute(HttpServletRequest request, HttpServletResponse response) {

String id = request.getParameter("id");

String pw = request.getParameter("pw");

String name = request.getParameter("name");

String email = request.getParameter("email");

String bbirth = request.getParameter("bbirth");

Timestamp birth = **null**;

**if**(bbirth!=**null**)

birth = Timestamp.*valueOf*(bbirth+" 00:00:00");

String address = request.getParameter("address");

MemberDto dto = **new** MemberDto(id, pw, name, email, birth, **null**, address);

MemberDao mDao = MemberDao.*getInstance*();

**int** joinResult = mDao.insertMember(dto);

request.setAttribute("joinResult", joinResult);

**return** **null**;

}

}

**MembersAllService.java**

**package** com.tj.ex;

**import** java.util.ArrayList;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** MembersAllService **implements** Service {

@Override

**public** ArrayList<MemberDto> execute(HttpServletRequest request, HttpServletResponse response) {

MemberDao dao = MemberDao.*getInstance*();

**return** dao.getMembersAll();

}

}

**FrontController.java**

**package** com.tj.ex;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.ArrayList;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("\*.do")

**public** **class** FrontController **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

actionDo(request, response);

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

request.setCharacterEncoding("utf-8");

actionDo(request, response);

}

**private** **void** actionDo(HttpServletRequest request, HttpServletResponse response) **throws** IOException {

String uri = request.getRequestURI(); // "/ch18/insert.do"

String conPath = request.getContextPath(); // "/ch18"

String command = uri.substring(conPath.length()); // "/insert.do"

String done = "";

Service service;

**if** (command.equals("/insert.do")) {

System.***out***.println("insert command 호출");

done = "insert command 호출";

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = " + uri + "<br>");

writer.println("conPath = " + conPath + "<br>");

writer.println("들어온 요청은 " + command + "<br>");

writer.println("진짜 " + done + "했다<br>");

writer.println("</body></html>");

writer.close();

} **else** **if** (command.equals("/select.do")) {

System.***out***.println("select command 호출");

done = "select command 호출";

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = " + uri + "<br>");

writer.println("conPath = " + conPath + "<br>");

writer.println("들어온 요청은 " + command + "<br>");

writer.println("진짜 " + done + "했다<br>");

writer.println("</body></html>");

writer.close();

} **else** **if** (command.equals("/update.do")) {

System.***out***.println("update command 호출");

done = "update command 호출";

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = " + uri + "<br>");

writer.println("conPath = " + conPath + "<br>");

writer.println("들어온 요청은 " + command + "<br>");

writer.println("진짜 " + done + "했다<br>");

writer.println("</body></html>");

writer.close();

} **else** **if** (command.equals("/delete.do")) {

System.***out***.println("delete command 호출");

done = "delete command 호출";

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("uri = " + uri + "<br>");

writer.println("conPath = " + conPath + "<br>");

writer.println("들어온 요청은 " + command + "<br>");

writer.println("진짜 " + done + "했다<br>");

writer.println("</body></html>");

writer.close();

} **else** **if** (command.equals("/memberJoin.do")) {

service = **new** MemberJoinService();

service.execute(request, response);

**int** result = (Integer) request.getAttribute("joinResult");

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body><h1>");

**if** (result == 0) {

writer.println("<h1>회원가입이 실패되었습니다</h1>");

} **else** {

writer.println("<h1>회원 가입되었습니다</h1>");

}

writer.println("</body><html>");

writer.close();

} **else** **if** (command.equals("/membersAll.do")) {

service = **new** MembersAllService();

ArrayList<MemberDto> members = service.execute(request, response);

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = response.getWriter();

writer.println("<html><body>");

writer.println("<table>");

**for** (MemberDto m : members) {

writer.println("<tr><td>" + m.getId() + "</td>");

writer.println("<td>" + m.getName() + "</td>");

writer.println("<td>" + m.getEmail() + "</td>");

writer.println("<td>" + m.getBirth() + "</td>");

writer.println("<td>" + m.getAddress() + "</td>");

writer.println("<td>" + m.getrDate() + "</td></tr>");

}

writer.println("</table></body></html>");

writer.close();

}

}

}

**Join.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

<script src=*"*${conPath }*/js/join.js"*></script>

</head>

<body>

<form action=*"*${conPath }*/memberJoin.do"* method=*"post"* name=*"join\_frm"*>

<table>

<caption>회원가입</caption>

<tr><th>아이디</th>

<td><input type=*"text"* name=*"id"*></td>

</tr>

<tr><th>비밀번호</th>

<td><input type=*"password"* name=*"pw"*></td>

</tr>

<tr><th>비밀번호</th>

<td><input type=*"password"* name=*"pwChk"*></td>

</tr>

<tr><th>이름</th>

<td><input type=*"text"* name=*"name"*></td>

</tr>

<tr><th>메일주소</th>

<td><input type=*"email"* name=*"email"* ></td>

</tr>

<tr><th>생년월일</th>

<td><input type=*"date"* name=*"birth"*></td>

</tr>

<tr><th>주소</th>

<td><input type=*"text"* name=*"address"*></td>

</tr>

<tr><td colspan=*"2"*>

<input type=*"button"* value=*"회원가입"* onclick="infoConfirm()">

<input type=*"reset"* value=*"취소"*>

<input type=*"button"* value=*"로그인"* onclick="location='login.jsp'">

</td>

</tr>

</table>

</form>

</body>

</html>몇일전에 했던 join.jsp에서 action부분만 변경. css폴더와 js폴더도 옮기기

**Members.sql**

**DROP** **TABLE** MEMBER **CASCADE** CONSTRAINTS;

**CREATE** **TABLE** MEMBER(

id VARCHAR2(30) **PRIMARY** **KEY**, -- ID

pw VARCHAR2(30) **NOT** **NULL**, -- PW

name VARCHAR2(30) **NOT** **NULL**, -- 이름

email VARCHAR2(30), -- 메일

birth **DATE**, -- 생년월일

rdate **DATE** **NOT** **NULL**, -- 가입한 날짜와 시간

address VARCHAR2(50)); -- 주소

**INSERT** **INTO** MEMBER (ID, PW, NAME, EMAIL, BIRTH, RDATE, ADDRESS)

**VALUES** ('bbb','111','구','hong@go.kr','1990-01-01', SYSDATE, '서울 종로');

**SELECT** \* **FROM** MEMBER;