**[ 18 ] Front Controller와 Command 구조**

여러분이 실무에서 작업을 할 때 큰 구조에서 두 패턴 구조를 필히 알고 있어야 다른 팀원과 co-work하는데 문제가 없고 개발시에도 효율적으로 개발을 할 수 있다. 난잡한 개발에서 체계화된 개발론으로도 볼 수 있다.

1. url-pattern
   1. 디렉터리 구조 ; 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/ch18/EmpList> → /EmpList 서블릿

http://localhost:8181/ch18/servlet/com.lec.ex.EmpList 서블릿의 긴 주소를 매핑한다.

* + 1. <http://localhost:8181/ch18/EmpSch> → /EmpSch 서블릿
  1. 확장자 구조 ; 확장자 형태로 서버의 해당 컴포넌트를 찾아서 실행하는 구조
     1. <http://localhost:8181/ch18/EmpList.do> ↘
     2. <http://localhost:8181/ch18/EmpSch.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 서블릿, JSP

요청2 처리 컴포넌트

요청3 처리 컴포넌트

DAO

요청1

요청2

요청3

**모든 요청을**

**처리하는 서블릿**

**= frontController**

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>

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

</head>

<body>

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

<h1><a href=*"http://localhost:8090/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** ServletException, IOException {

String uri = request.getRequestURI();

StringBuffer url = request.getRequestURL();

String conPath = request.getContextPath();

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

System.***out***.println("url = "+url);

System.***out***.println("uri = "+uri);

System.***out***.println("conPath = "+conPath);

System.***out***.println("command = "+command);

String viewPage = **null**;

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

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

request.setAttribute("command", "insert command 호출");

viewPage = "ex2/insert.jsp";

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

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

request.setAttribute("command", "select command 호출");

viewPage = "ex2/select.jsp";

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

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

request.setAttribute("command", "update command 호출");

viewPage = "ex2/update.jsp";

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

System.***out***.println("deletecommand 호출");

request.setAttribute("command", "deletecommand 호출");

viewPage = "ex2/delete.jsp";

}

RequestDispatcher dispatcher = request.getRequestDispatcher(viewPage);

dispatcher.forward(request, response);

}

}

**ex2/insert.jsp**

<body>

<h2>들어온 요청 결과는 ${command }입니다</h2>

</body>

</html>

**ex2/select.jsp**

<body>

<h2>${result }입니다</h2>

<h3>select결과는 ${list}</h3>

</body>

</html>

**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** ServletException, IOException {

//response.getWriter().append("<h1>FrontController</h1>");

String uri = request.getRequestURI();

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

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

System.***out***.println("uri = "+uri);

System.***out***.println("conPath = "+conPath);

System.***out***.println("들어온 요청은 "+command);

String viewPage = **null**;

Service service = **null**;

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

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

// insert 로직

request.setAttribute("result", "삽입 했음");

viewPage = "select.do";

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

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

// select 로직

request.setAttribute("list", "셀렉트 한 결과");

viewPage = "ex2/select.jsp";

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

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

// delete 로직

request.setAttribute("result", "딜리트 했음");

viewPage = "select.do";

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

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

request.setAttribute("result", "업데이트 했음");

viewPage = "select.do";

}

RequestDispatcher dispatcher = request.getRequestDispatcher(viewPage);

dispatcher.forward(request, response);

}

}

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

insert.do FrontController.java InsertService.java

delete.do

직접 처리하지 않고 command클래스를 하나씩 만들어 준다

**FrontController.java**

**import** com.lec.service.\*;

@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** ServletException, IOException {

//response.getWriter().append("<h1>FrontController</h1>");

String uri = request.getRequestURI();

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

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

System.***out***.println("uri = "+uri);

System.***out***.println("conPath = "+conPath);

System.***out***.println("들어온 요청은 "+command);

String viewPage = **null**;

Service service = **null**;

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

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

// insert 로직

service = **new** InsertService();

service.execute(request, response);

viewPage = "select.do";

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

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

// select 로직

service = **new** SelectService();

service.execute(request, response);

viewPage = "ex2/select.jsp";

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

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

// delete 로직

service = **new** DeleteService();

service.execute(request, response);

viewPage = "select.do";

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

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

service = **new** UpdateService();

service.execute(request, response);

viewPage = "select.do";

}

RequestDispatcher dispatcher = request.getRequestDispatcher(viewPage);

dispatcher.forward(request, response);

}

}

**Service.java**

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

**public** **void** execute(HttpServletRequest request, HttpServletResponse response);

}

**DeleteService.java**

**public** **class** DeleteService **implements** Service{

@Override

**public** **void** execute(HttpServletRequest request, HttpServletResponse response) {

**int** deleteResult = 1;

request.setAttribute("deleteResult", deleteResult);

}

}

**InsertService.java**

**public** **class** InsertService **implements** Service{

@Override

**public** **void** execute(HttpServletRequest request, HttpServletResponse response) {

request.setAttribute("insertResult", 1);

}

}

**SelectService.java**

**public** **class** SelectService **implements** Service{

@Override

**public** **void** execute(HttpServletRequest request, HttpServletResponse response) {

request.setAttribute("list", "list들 결과");

}

}

**updateService.java**

**public** **class** UpdateService **implements** Service{

@Override

**public** **void** execute(HttpServletRequest request, HttpServletResponse response) {

request.setAttribute("updateResult", 1);

}

}

ex2/select.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>

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

</head>

<body>

<c:if test=*"*${insertResult **eq** 1}*"*>

<script>alert('입력 성공');</script>

</c:if>

<c:if test=*"*${deleteResult **eq** 1}*"*>

<script>alert('삭제 성공');</script>

</c:if>

<c:if test=*"*${updateResult **eq** 1}*"*>

<script>alert('수정 성공');</script>

</c:if>

<h1>select.jsp입니다</h1>

<hr>

<h2>select 결과 : ${list }</h2>

</body>

</html>