

```

1 1. Get / Set Method
2 1)getset.html
3
4 <!DOCTYPE html>
5 <html lang='ko'>
6 <head>
7 <meta charset="EUC-KR">
8 <title>GET 과 POST 테스트</title>
9 </head>
10 <body>
11 <h1>도서 구매 정보 입력</h1>
12 <h4>각 사항들에 대해 입력해 주세요</h4>
13 <hr>
14 <form method="get" action="/DemoWeb/QueryGetServlet">
15 이름 : <input type="text" name="buyername"><br>
16 주소 : <input type="text" name="buyeraddr" size="30"><br>
17 <strong>선택 도서명 : </strong><br>
18 <input type="checkbox" name="buyeritem" value="Java Programming">Java Programming<br>
19 <input type="checkbox" name="buyeritem" value="Servlet JSP Programming">Servlet JSP
20 Programming<br>
21 <input type="checkbox" name="buyeritem" value="JDBC Programming">JDBC Programming<br>
22 <p>
23 <input type="submit" value="주문"><input type="reset" value="다시 입력">
24 </p>
25 </form>
26 </body>
27 </html>

```

## 2)QueryGetServlet.java

```

28
29
30 public class QueryGetServlet extends HttpServlet {
31     protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException,
32         IOException {
33         res.setContentType("text/html;charset=utf-8");
34         PrintWriter out = res.getWriter();
35         //req.setCharacterEncoding("utf-8");
36         String name = entoko(req.getParameter("buyername"));
37         String addr = entoko(req.getParameter("buyeraddr"));
38         String [] item = req.getParameterValues("buyeritem");
39         out.println("<html><body>");
40         out.println("<font color='red'><h2>");
41         out.println("<input type='checkbox' name='buyeritem' value='Java Programming'>Java Programming");
42         out.println("<input type='checkbox' name='buyeritem' value='Servlet JSP Programming'>Servlet JSP");
43         out.println("<input type='checkbox' name='buyeritem' value='JDBC Programming'>JDBC Programming");
44         out.println("<p>");
45         out.println("<input type='submit' value='주문'><input type='reset' value='다시 입력'>");
46         out.println("</p>");
47         out.println("</form>");
48         out.println("</body>");
49         out.println("</html>");
50         out.close();
51     }
52     private String entoutf8(String en){
53         String utf8 = null;
54         try{
55             utf8 = new String(en.getBytes("ISO8859_1"), "UTF-8");
56         }catch(IOException ex){}
57         return utf8;
58     }
59 }

```

## 2. Cookie

### 1)CookieSetServlet.java

```

60
61
62 public class CookieSetServlet extends HttpServlet {
63     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
64         ServletException, IOException {
65         Cookie c1 = new Cookie("discount", "20");

```

```

72     Cookie c2 = new Cookie("item", "sports");
73     response.setContentType("text/html;charset=utf-8");
74     response.addCookie(c1);
75     c2.setMaxAge(60 * 60 * 24 * 3);
76     response.addCookie(c2);
77     PrintWriter out = response.getWriter();
78     out.println("<html><body>");
79     out.println("<h1>Cookie 설정</h1><hr>");
80     out.println("설정 완료!!!");
81     out.println("</body></html>");
82     out.close();
83 }
84 }

```

## 2)CookieGetServlet.java

```

87
88 public class CookieGetServlet extends HttpServlet {
89     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
90         response.setContentType("text/html;charset=utf-8");
91         PrintWriter out = response.getWriter();
92         out.println("<html><body>");
93         out.println("<h1>Cookie 추출</h1><hr>");
94         Cookie [] cookies = request.getCookies();
95         if(cookies != null && cookies.length > 0){
96             out.println("다음과 같은 Cookie 정보가 전달 되었습니다.<br>");
97             for(int i = 0 ; i < cookies.length; i++){
98                 out.print("Cookie Name : " + cookies[i].getName() + "<br>");
99                 out.println("Cookie Value : " + cookies[i].getValue() + "<br>");
100             }
101         }else
102             out.println("전달된 cookie 정보가 없습니다.");
103         out.println("</body></html>");
104         out.close();
105     }
106 }

```

## 3. Session

### 1)SessionTestServlet.java

```

112 public class SessionTestServlet extends HttpServlet {
113     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
114         response.setContentType("text/html;charset=utf-8");
115         PrintWriter out = response.getWriter();
116         HttpSession session = null;
117         String msg = null;
118         String param = request.getParameter("comm");
119         if(param.equals("create")){
120             session = request.getSession();
121             if(session.isNew())
122                 msg = "세션 객체가 새로 생성됐습니다";
123             else
124                 msg = "이미 세션 객체가 생성되어 있습니다";
125         }else if(param.equals("destroy")){
126             session = request.getSession(false);
127             if(session != null){
128                 session.invalidate();
129                 msg = "세션 객체가 삭제됐습니다.";
130             }else
131                 msg = "전달된 명령을 해석할 수 없습니다";
132         }
133         out.println("<html><body><h3>처리결과 : </h3>" + msg);
134         out.println("</body></html>");
135         out.close();
136     }
137 }

```

### 2)CountTestServlet.java

```

141 public class CountTestServlet extends HttpServlet {
142     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

```

```

143         response.setContentType("text/html;charset=utf-8");
144         PrintWriter out = response.getWriter();
145         out.println("<html><body bgcolor='yellow'");
146         out.println("<h2>누적시킬 숫자를 입력하세요...</h2>");
147         out.println("<form method='post' action='\"/DemoWeb/CountTestServlet\">");
148         out.println("<p><input type='\"text\"' name='\"number\"'></p>");
149         out.println("<input type='\"submit\"' name='\"see\"' value='\"전송\"'>");
150         out.println("<input type='\"submit\"' name='\"buy\"' value='\"재입력\"'>");
151         out.println("</form></body></html>");
152         out.close();
153     }
154     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
155         HttpSession session = request.getSession(true);
156         if(session.isNew())
157             session.setAttribute("number", new int[]{0});
158         int [] sum = (int [])session.getAttribute("number");
159         int number = Integer.parseInt(request.getParameter("number"));
160         sum[0] += number;
161         response.setContentType("text/html;charset=euc-kr");
162         PrintWriter out = response.getWriter();
163         out.println("<html><body><h3>누적된 값 : " + sum[0] + "</h3>");
164         out.println("<h3>세션 ID : " + session.getId() + "</h3>");
165         out.println("<h3>세션발생시간 : " + new java.util.Date(session.getCreationTime()).toString() + "</h3>");
166         out.println("<hr><a href='\"" + request.getRequestURI() + "\">이전페이지</a>");
167         out.println("</body></html>");
168         out.close();
169     }
170 }
171
172

```

#### 4. Include

##### 1)IncludeTestServlet1.java

```

175
176     public class IncludeTestServlet1 extends HttpServlet {
177         protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException {
178             res.setContentType("text/html;charset=utf-8");
179             PrintWriter out = res.getWriter();
180             out.println("<html><head><title>Include 예제</title></head>");
181             out.println("<body><h3>IncludeTestServlet1의 페이지입니다.</h3></body></html>");
182             ServletContext context = getServletContext();
183             RequestDispatcher rd = context.getRequestDispatcher("/IncludeTestServlet2");
184             out.println("<hr color='red'>");
185             rd.include(req, res);
186             out.println("<hr color='blue'>");
187             rd.include(req, res);
188             out.close();
189         }
190     }
191

```

##### 2)IncludeTestServlet2.java

```

192
193
194     public class IncludeTestServlet2 extends HttpServlet {
195         protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException {
196             res.setContentType("text/html;charset=utf-8");
197             PrintWriter out = res.getWriter();
198             out.println("<h3>IncludeTestServlet2의 페이지입니다.</h3>");
199         }
200     }
201

```

##### 3)InServlet.java

```

202
203
204     public class InServlet extends HttpServlet {
205         public void doGet(HttpServletRequest req, HttpServletResponse res)
206             throws IOException, ServletException {
207             res.setContentType("text/html;charset=utf-8");
208             PrintWriter out = res.getWriter();
209             out.println("안녕하세요.");
210             out.println("내부 서블릿입니다.");
211             //out.close();   ***주의 닫지 말 것
212         }
213     }

```

#### 4)IncludingServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class IncludingServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws IOException, ServletException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();

        out.println("<html><head><title>");
        out.println("Including Servlet");
        out.println("</title></head><body bgcolor='yellow'>");
        out.println("<center><h2>Including Servlet</h2></center>");
        out.println("다음 내용은 다른 서블릿의 내용을");
        out.println("include한 것입니다. <hr>");

        ServletContext sc = getServletContext();
        RequestDispatcher rd = sc.getRequestDispatcher("/servlet/In");
        rd.include(req, res);

        out.println("<hr>이곳은 다시 IncludingServlet 입니다.");
        out.println("</body></html>");
    }
}
```

#### 5)myimage.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert title here</title>
</head>
<body>
    <h1 style='text-align:center'>여신</h1>
    <div style='text-align:center'></div>
</body>
</html>
```

#### 6)IncludeDemoServlet.java

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

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class IncludeDemoServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html lang='ko'>");
        out.println("<body bgcolor='red'>");
        ServletContext context = this.getServletContext();
        RequestDispatcher rd =
            context.getRequestDispatcher("/myimage.html");
        rd.include(req, res);
        out.println("<hr>");
        rd = context.getRequestDispatcher("/servlet/Visit");
        rd.include(req, res);
        out.println("</body></html>");
        out.close();
    }
}
```

```

288     }
289     @Override
290     protected void doPost(HttpServletRequest req, HttpServletResponse res)
291         throws ServletException, IOException{
292         this.doGet(req, res);
293     }
294 }
295
296

```

## 297 5. Forward & Redirect

298 javax.servlet.Request Dispatcher (요청재전송) --> Forwarding

- 299 1)사용자의 요청을 받은 서블릿 혹은 JSP 에서 다른 컴포넌트(다른 서블릿, 다른 JSP, html)로 요청을 위임할 수 있는 방법
- 300 2)사용이유는 1)보안, 2) 처리작업의 모듈화
- 301 3)직접 요청을 받은 서블릿 혹은 JSP는 모든 작업을 직접 처리하지 않고 모듈화시킨 다른 컴포넌트로 요청을 위임하여 처리할 수 있는 방법
- 302 4)재사용성이 높아지고, 유지보수가 쉬워짐
- 303 5)일반적으로 MVC패턴에서 서블릿이 JSP 페이지로 포워딩할 때 주로 사용함.
- 304 6)사용자로부터 요청을 받은 페이지와 사용자에게 응답을 처리하는 페이지를 구별할 수 있다.
- 305 7)일반적으로 이러한 패턴을 FrontController 패턴이라고 한다.
- 306 8)요청을 받는 페이지는 서블릿으로 구현하고, 응답을 처리하는 페이지는 JSP 로 구현할 수 있다.
- 307 9)Model2(MVC) 의 핵심기능이다.
- 308 10)요청한 자원의 포워딩 방법 2가지
  - 309 -RequestDispatcher의 forward() 메소드
  - 310 -ServletResponse 의 sendRedirect() 메소드

## 311 11)Lab

312 -product.html

```

313
314 <!DOCTYPE html>
315 <html>
316 <head>
317 <meta charset="utf-8">
318 <title>상품페이지</title>
319 </head>
320 <body>
321 <h1>상품 목록</h1>
322 <form action="/0613/servlet/CartSave" method="post">
323 <input type="radio" name="product" value="Computer">Computer<br />
324 <input type="radio" name="product" value="Sonata">Sonata<br />
325 <input type="radio" name="product" value="Keyboard">Keyboard<br />
326 <input type="radio" name="product" value="Book">Book<br />
327 <input type="submit" value="답기">
328 </form>
329 </body>
330 </html>
331

```

332 -ForwardDemoServlet.java

```

333
334 import java.io.IOException;
335
336 import javax.servlet.RequestDispatcher;
337 import javax.servlet.ServletContext;
338 import javax.servlet.ServletException;
339 import javax.servlet.http.HttpServlet;
340 import javax.servlet.http.HttpServletRequest;
341 import javax.servlet.http.HttpServletResponse;
342
343 public class ForwardDemoServlet extends HttpServlet {
344     @Override
345     protected void doGet(HttpServletRequest req, HttpServletResponse res)
346         throws ServletException, IOException{
347         ServletContext context = this.getServletContext();
348         RequestDispatcher rd =
349             context.getRequestDispatcher("/product.html");
350         rd.forward(req, res);
351     }
352     @Override
353     protected void doPost(HttpServletRequest req, HttpServletResponse res)
354         throws ServletException, IOException{
355         this.doGet(req, res);
356     }
357 }
358

```

359 -ForwardTestServlet1.java

```

362 import java.io.*;
363 import javax.servlet.*;
364 import javax.servlet.http.*;
365 public class ForwardTestServlet1 extends HttpServlet {
366     protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException,
        IOException {
367         res.setContentType("text/html;charset=utf-8");
368         PrintWriter out = res.getWriter();
369         out.println("<html><head><title>Forward 예제");
370         out.println("</title></head><body bgcolor='red'><h3>ForwardTestServlet1 의 수행결과 </h3>");
371         out.println("</body></html>");
372         ServletContext context = getServletContext();
373         RequestDispatcher rd = context.getRequestDispatcher("/servlet/ForwardTestServlet2");
374         rd.forward(req,res);
375         out.close();
376     }
377 }
378 }
379
380 -ForwardTestServlet2.java
381
382 public class ForwardTestServlet2 extends HttpServlet {
383     protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException,
        IOException {
384         res.setContentType("text/html;charset=utf-8");
385         PrintWriter out = res.getWriter();
386         out.println("<html><head><title>Forward 예제");
387         out.println("</title></head><body bgcolor='yellow'><h3>나는 ForwardTestServlet2 입니다.</h3>");
388         out.println("</body></html>");
389         out.close();
390     }
391 }
392

```

## 12)Redirect

- Forward와 마찬가지로 다른 웹 컴포넌트에게 페이지를 위임하는 방법
- 차이점은 응답을 먼저하고 클라이언트의 브라우저가 다시 재요청하기 때문에, 브라우저의 URL 과 새로 응답하는 페이지가 같다.
- response.sendRedirect(target)
- target은 동일 컨텍스트의 서브릿, 혹은 JSP, 혹은 Html, 다른 컨텍스트, 다른 도메인, 다른 웹 서버도 가능

### -RedirectDemoServlet.java

```

400
401 import java.io.IOException;
402 import java.io.PrintWriter;
403
404 import javax.servlet.ServletException;
405 import javax.servlet.http.HttpServlet;
406 import javax.servlet.http.HttpServletRequest;
407 import javax.servlet.http.HttpServletResponse;
408
409 public class RedirectDemoServlet extends HttpServlet {
410     @Override
411     protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
412         res.setContentType("text/html;charset=utf-8");
413         PrintWriter out = res.getWriter();
414         out.println("<!DOCTYPE html>");
415         out.println("<html lang='ko'>");
416         out.println("<body bgcolor='red'>");
417         //res.sendRedirect("http://www.javaexpert.co.kr"); 1. 첫번째 방법
418         out.println("<script>"); //2. 두번째 방법
419         out.println("location.href = 'http://www.naver.com';");
420         out.println("</script>");
421         out.println("</body></html>");
422         out.close();
423     }
424 }
425 @Override
426 protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
427     this.doGet(req, res);
428 }
429 }
430 }
431
432
433

```

```

434 5. ServletContext API
435 1)ServletContext 는 웹 어플리케이션(Context)마다 하나씩 생성되는 객체
436 2)웹 어플리케이션의 LifeCycle 과 일치
437 3)application scope
438
439
440 6. Servlet/JSP 의 4가지 Scope
441 1)Application Scope
442 -웹 어플리케이션과 LifeCycle 이 일치한다.
443 -웹 어플리케이션이 존재하는 동안에는 모든 서블릿과 JSP 에서 접근 가능.
444 -ServletContext 객체
445
446 2)Session Scope
447 -브라우저의 LifeCycle과 동일한 Scope 를 의미
448 -브라우저를 종료하지 않으면 서블릿과 JSP 에서 접근 가능
449 -일반적으로 time-out 방법을 사용한다.
450 -HttpSession 객체
451
452 3)Request Scope
453 -브라우저에게 요청하고 서버에서 응답하는 Scope 까지를 의미
454 -사용자 입장에서 요청한 화면이 보여질 때까지의 범위
455 -HttpServletRequest 해당
456
457 4)Page Scope
458 -현재 브라우저에 보여지는 웹 페이지 자체
459 -자바의 this
460
461
462 7. ServletContext 의 핵심 메소드
463 String getInitParameter(name)
464 InputStream getResourceAsStream(path)
465 void setAttribute(name, value)
466 Object getAttribute(name)
467
468
469 8. Context Parameter 설정
470 1)ContextParamServlet.java
471
472 @WebServlet("/ContextParam")
473 public class ContextParamServlet extends HttpServlet {
474     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
475     IOException {
476         String driver = getServletContext().getInitParameter("driver");
477         String savePath = getServletContext().getInitParameter("savePath");
478
479         response.setContentType("text/html; charset=EUC-KR");
480         PrintWriter out = response.getWriter();
481         out.print("<html><body>");
482         out.print("드라이버명: " + driver + "<br>");
483         out.print("저장 경로: " + savePath + "<br>");
484         out.print("</body></html>");
485     }
486 }
487
488 2)web.xml
489
490 <context-param>
491     <param-name>driver</param-name>
492     <param-value>oracle.jdbc.driver.OracleDriver</param-value>
493 </context-param>
494 <context-param>
495     <param-name>savePath</param-name>
496     <param-value>c:\\save</param-value>
497 </context-param>
498
499 9. 서블릿에서 파일 접근
500 -읽기 모드만 가능
501
502 1)testFile.txt
503 안녕하세요.
504 ServletContext 객체를 이용한 파일읽기입니다.
505 감사합니다.
506

```

## 2)ContextFileServlet.java

```
@WebServlet("/ContextFile")
public class ContextFileServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {

        String readFile = "/WEB-INF/testFile.txt";
        InputStream is = getServletContext().getResourceAsStream(readFile);
        BufferedReader reader = new BufferedReader(new InputStreamReader(is));

        response.setContentType("text/html; charset=EUC-KR");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");

        String str = reader.readLine();
        while(str != null){
            out.print(str + "<br>");
            str = reader.readLine();
        }
        reader.close();
        out.print("</body></html>");
    }
}
```

## 10. 서블릿에서 속성 설정 및 참조

### 1)ContextSetServlet.java

```
@WebServlet("/ContextSet")
public class ContextSetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        // 속성값 설정
        String name = "홍길동";
        int age = 20;

        getServletContext().setAttribute("name", name );
        getServletContext().setAttribute("age", age );
    }
}
```

### 2)ContextGetServlet.java

```
@WebServlet("/ContextGet")
public class ContextGetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        //속성값 얻기
        String name = (String)getServletContext().getAttribute("name");
        int age = (Integer)getServletContext().getAttribute("age");

        response.setContentType("text/html; charset=EUC-KR");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");
        out.print("이름 : " + name + "<br>");
        out.print("나이 : " + age + "<br>");
        out.print("</body></html>");
    }
}
```

## 11. ServletContextListener API

-웹 어플리케이션이 초기화되고 제거되는 이벤트를 감지하는 ServletContextListener API를 사용하면, 언제 초기화되고 제거 되었는지를 쉽게 알 수 있다.

-이 이벤트는 JDBC의 Pooling 기법에 적용 가능

-웹 어플리케이션이 초기화될 때 Pooling 을 활성화하고, 제거될 때 Pooling 을 비활성화 시키면 효율적으로 Connection 을 관리가능

-순서

1)ServletContextListener interface를 구현한 클래스 작성

2)web.xml 에 구현한 클래스를 <listener> 태그로 등록

또는 @WebListener 어노테이션 사용

3)Tomcat 컨테이너 시작, 종료를 통해 이벤트 감지 확인

-Lab

1)ContextListenerImpl.java

```
public class ContextListenerImpl implements ServletContextListener {
```



```

577         @Override
578         public void contextDestroyed(ServletContextEvent event) {
579             System.out.println("웹 어플리케이션 제거");
580         }
581         @Override
582         public void contextInitialized(ServletContextEvent event) {
583             System.out.println("웹 어플리케이션 초기화");
584         }
585     }
586
587 2)web.xml
588     <listener>
589         <listener-class>kr.co.javaexpert.libs.ContextListenerImpl</listener-class>
590     </listener>
591
592 3)ContextListenerImpl.java
593     @WebListener <----@WebListener 를 이용한 리스너 처리
594     public class ContextListenerImpl implements ServletContextListener {
595         @Override
596         public void contextDestroyed(ServletContextEvent event) {
597             System.out.println("웹 어플리케이션 제거");
598         }
599         @Override
600         public void contextInitialized(ServletContextEvent event) {
601             System.out.println("웹 어플리케이션 초기화");
602         }
603     }

```