

서블릿의 한글 처리와 데이터 통신

서블릿에서 응답시 한글 처리

❖ 서블릿

- 패키지 : unit03
- 클래스명 : HelloServlet03

```
@WebServlet("/HelloServlet03")
public class HelloServlet03 extends HttpServlet {
    ...
    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.print("<html><body><h1>");
        out.print("Hello Servlet");
        out.print(" 헬로우 서블릿");
        out.print("</h1></body></html>");
        out.close();
    }
}
```

서블릿에서 응답시 한글 처리

❖ 서블릿

- 한글 문자셋 설정
 - UTF-8

```
@WebServlet("/HelloServlet03")
public class HelloServlet03 extends HttpServlet {
    ...
    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.print("<html><body><h1>");
        out.print("Hello Servlet");
        out.print(" 헬로우 서블릿");
        out.print("</h1></body></html>");
        out.close();
    }
}
```

서블릿에서 응답시 한글 처리

❖ 요청에서 한글 처리

- 요청 정보
 - <form>태그를 통해 전달
- GET/POST 방식에 따라 처리가 한글 처리가 달라짐

```
<form method="get/post" action="호출할서블릿">
```

속성	설명
method	어떤 방식으로 데이터를 넘겨 줄 것인지를 결정한다. get과 post 방식 중에서 하나를 선택한다. post는 사용자가 입력한 내용을 발송할 때 그 내용을 공개하지 않는다. get은 사용자가 입력한 결과를 알 수 있도록 내용을 공개한다.
action	전송(submit) 버튼을 누르면 action 속성 다음에 기술한 URL에 지정된 파일로 이동한다.

- get 방식으로 전송한 경우 해당 서블릿의 doGet() 메서드 호출
- post 방식으로 전송한 경우 해당 서블릿의 doPost() 메서드 호출

서블릿에서 응답시 한글 처리

❖ 요청에서 한글 처리

- GET 방식으로 요청하기

```
<form method="get" action="MethodServlet">
```

- form 데이터의 전송
 - submit 버튼 클릭시 전송

```
<input type="submit" value="전송">
```

- form 데이터 입력 취소
 - reset 버튼 클릭시 form의 초기 데이터로 리셋
 - 사용자 입력 데이터 사라짐

```
<input type="reset" value="취소">
```

서블릿에서 응답시 한글 처리

❖ form 처리

- 04_method.jsp
 - get 방식 전송 버튼과 post 방식 전송 버튼 구성
- MethodServlet.java
 - 전송 방식에 따라 doGet, doPost 메서드 호출

서블릿에서 응답시 한글 처리

❖ 04_method.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>서블릿</title>
</head>
<body>
    <form method="get" action="MethodServlet">
        <input type="submit" value="get 방식으로 호출하기">
    </form>
    <br>
    <br>
    <form method="post" action="MethodServlet">
        <input type="submit" value="post 방식으로 호출하기">
    </form>
</body>
</html>
```

서블릿에서 응답시 한글 처리

❖ MethodServlet.java

```
package unit04;
:
@WebServlet("/MethodServlet")
public class MethodServlet extends HttpServlet {
:
    protected void doGet(HttpServletRequest request,
                           HttpServletResponse response)
                           throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.print("<h1>get 방식으로 처리됨</h1>");
        out.close();
    }
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
                           throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.print("<h1>post 방식으로 처리됨</h1>");
        out.close();
    }
}
```


쿼리 스트링

❖ 사용자 입력 form 데이터의 전송

○ 쿼리 스트링

- get 방식으로 클라이언트 서버로 데이터를 전송하는 방법
- "리소스(요청url)?이름1=값1&이름2=값2..." 형태를 가짐

ParamServlet? id = pinksung & age = 15

이름 값 이름 값

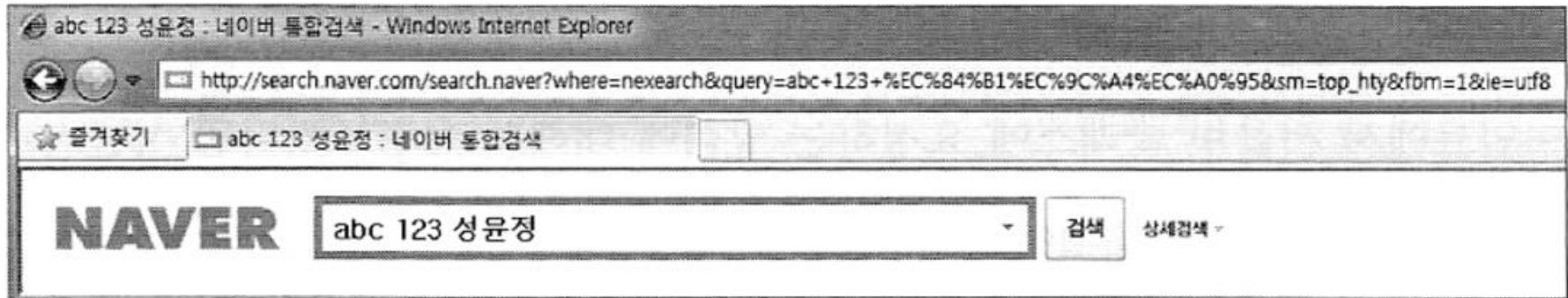
사용자가 입력한 값(pinksung)을 서버에서 얻어오려면 이름(id)을 알아야 합니다.

- 이름 : form요소의 name 속성으로 지정한 명칭
- 값 : 해당 요소에 사용자가 입력한 값 또는 value 속성으로 지정한 값

쿼리 스트링

❖ 사용자 입력 form 데이터의 전송

- 쿼리 스트링



http://search.naver.com/search.naver ? where=nexearch&query=abc+123+%EC%84%B1%EC%9C%A4%EC%A0%95&sm=top_h ty&fbm=1&ie=utf8

① URL

② 쿼리 스트링

where = nexearch & query = abc+123+%EC%84%B1%EC%9C%A4%EC%A0%95 & sm= top_h ty & fbm = 1 & ie = utf8

① 이름 값 ② 이름 값 ③ 이름 값 ④ 이름 값 ⑤ 이름 값

쿼리 스트링

❖ <form> 입력 요소

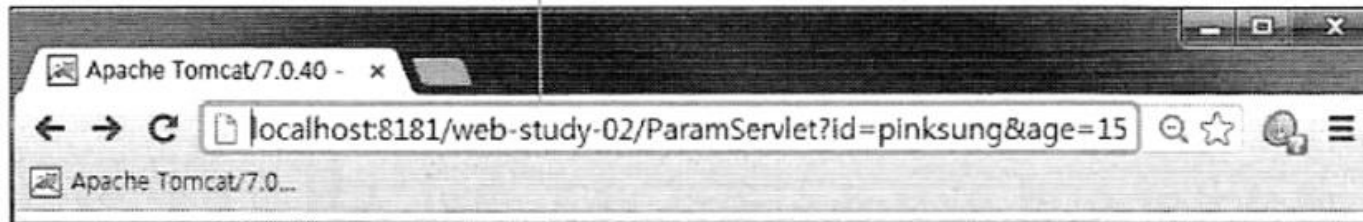
○ <input>

- 텍스트 박스
 - type="text"
 - 문자열 입력을 위해 사용

```
<input type="text" name="텍스트박스 이름">
```

```
<input type="text" name="id">  
<input type="text" name="age">
```

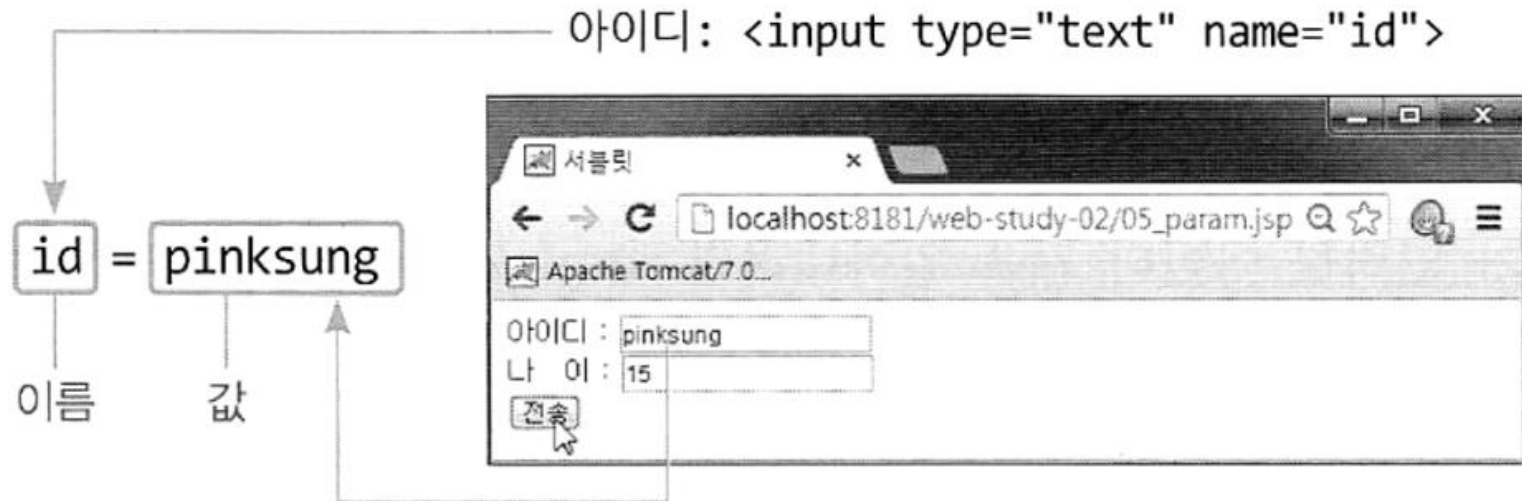
<http://localhost:8181/web-study-02/ParamServlet?id=pinksung&age=15>



쿼리 스트링

❖ <form> 입력 요소

○ <input>



요청 객체(request)와 파라미터 관련 메서드(getParameter)

❖ 서블릿에서 폼 요청 값 얻기

- request 객체의 getParameter() 메서드 호출
- 인자로 폼요소의 이름을 지정

```
String id = req.getParameter("id");
```

input 태그의 name 속성값

```
<input type="text" name="id">
```

```
String id=req.getParameter("id");
```

```
<input type="text" name="id">
```

이름 값

아이디 : pinksung
나 이 : 20
전송

요청 객체(request)와 파라미터 관련 메서드(getParameter)

❖ 서블릿에서 폼 요청 값 얻기

- request.getParameter() 메서드
 - 모든 값을 String 타입으로 리턴
 - 적절한 데이터 타입 변환 필요

```
int age=Integer.parseInt(request.getParameter("age"));
```

String형을 int형으로
변환하는 메소드

- 지정한 파라미터가 없는 경우 null 리턴
 - 이경우 형변환시 예외 발생
 - null인 경우 디폴트 값을 가지도록 처리 필요
 - 또는 클라이언트 쪽에서 입력을 반드시 하도록 검증

요청 객체(request)와 파라미터 관련 메서드(getParameter)

❖ 예제

파일 이름	설명
05_param.jsp	아이디와 나이를 입력 받는 폼 양식을 갖추고 있다. [전송] 버튼을 클릭하면 서블릿으로 입력된 값이 전송되고 서블릿에서 입력받은 값을 받아 처리한다.
ParamServlet.java	HTML 문서의 입력 양식에 입력한 내용을 처리하는 서블릿 클래스이다. 하이퍼링크 “다시”를 클릭하면 HTML 문서로 되돌아가서 새로운 데이터를 입력받는다.

❖ 05_param.jsp

16

요청 객체(request)와 파라미터 관련 메서드(getParameter)

❖ ParamServlet.java

```
package unit05;

:

@WebServlet("/ParamServlet")
public class ParamServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public ParamServlet() {
        super();
    }
}
```

요청 객체(request)와 파라미터 관련 메서드(getParameter)

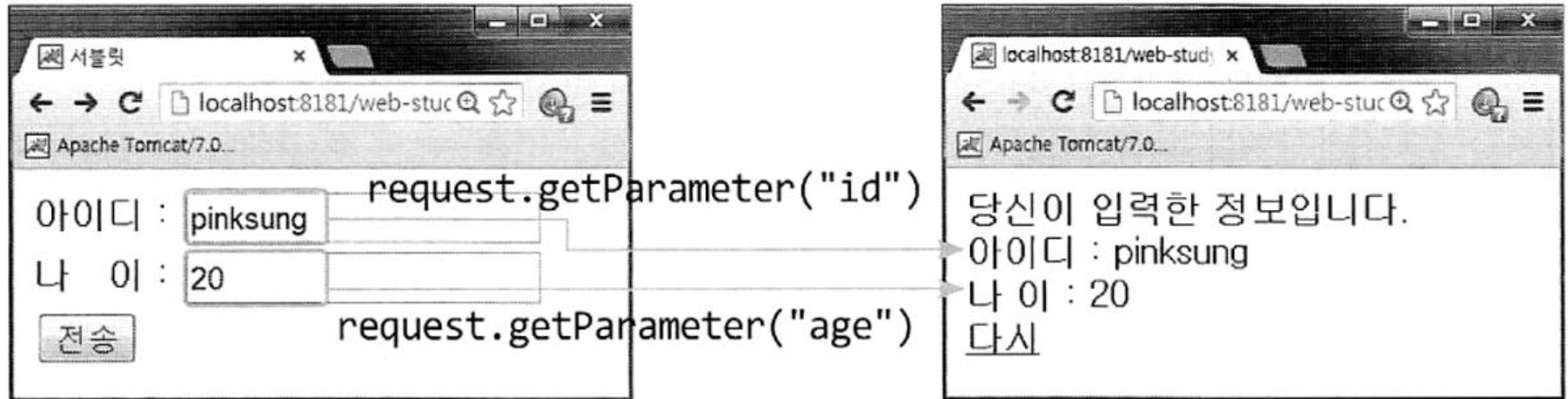
❖ ParamServlet.java

```
protected void doGet(HttpServletRequest request,
                    HttpServletResponse response)
                    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    String id = request.getParameter("id");
    int age = Integer.parseInt(request.getParameter("age"));
    PrintWriter out = response.getWriter();
    out.print("<html><body>");
    out.println("당신이 입력한 정보입니다.<br>");
    out.println("아 이 디 : ");
    out.println(id);
    out.println("<br> 나이 : ");
    out.println(age);
    //자바스크립트로 이전 페이지로 이동하는 링크를 만들어 줌
    out.println("<br><a href='javascript:history.go(-1)''>다시</a>");
    out.print("</body></html>");
    out.close();
}
```

요청 객체(request)와 파라미터 관련 메서드(getParameter)

❖ 예제



- 입력하지 않고 전송시
 - 에러 발생
 - id, age 모두 null 배정
 - null에 대해서 `Integer.parseInt()` 호출시 예외 발생
 - 브라우저에서 입력이 있는 경우에만 전송되도록 강제화 필요

요청 객체(request)와 파라미터 관련 메서드(getParameter)

❖ 05_param.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>서블릿</title>
    <script type="text/javascript" src="param.js"></script>
</head>
<body>
    <form method="get" action="ParamServlet" name="frm">
        아이디 : <input type="text" name="id"><br>
        나 &nbsp; ; 이 : <input type="text" name="age"><br>
        <input type="submit" value="전송" onclick="return check()">
    </form>
</body>
</html>
```

요청 객체(request)와 파라미터 관련 메서드(getParameter)

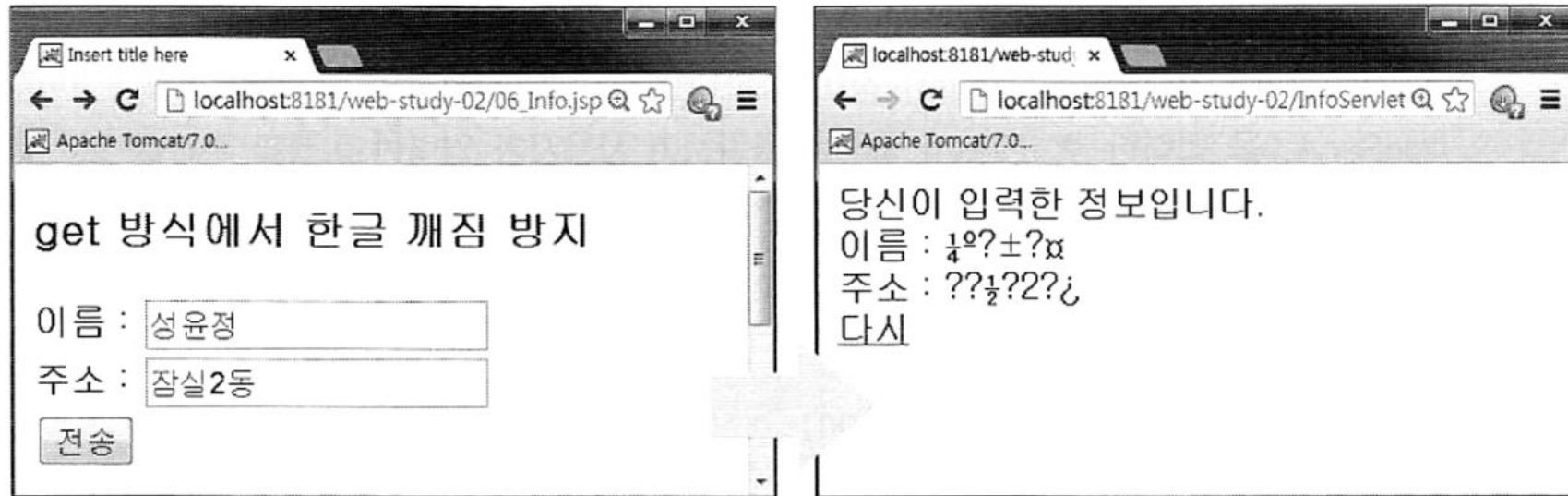
❖ param.js

```
function check() {  
    if (document.frm.id.value == "") {  
        alert("아이디를 입력해주세요.");  
        document.frm.id.focus();  
        return false;  
    } else if (document.frm.age.value == "") {  
        alert("나이를 입력해주세요.");  
        document.frm.age.focus();  
        return false;  
    } else if (isNaN(document.frm.age.value)) {  
        alert("숫자로 입력해주세요.");  
        document.frm.age.focus();  
        return false;  
    } else {  
        return true;  
    }  
}
```

- true를 리턴하면 전송
- false를 리턴하면 전송 취소

서블릿에서 한글 요청 처리

❖ 한글 요청



서블릿에서 한글 요청 처리

❖ 예제

파일 이름	설명
06_Info.jsp	이름과 주소를 입력 받는 폼 양식을 갖는다. [전송] 버튼을 클릭하면 서블릿으로 입력된 값이 전송되고 서블릿에서 입력받은 값을 받아 처리한다.
InfoServlet	HTML 문서에서 입력한 값을 읽어오는 서블릿 클래스이다.

서블릿에서 한글 요청 처리

❖ 06_Info.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
:
<body>
    <h3> get 방식에서 한글 깨짐 방지 </h3>
    <form method="get" action="InfoServlet">
        이름 : <input type="text" name="name"><br>
        주소 : <input type="text" name="addr"><br>
        <input type="submit" value="전송" >
    </form>

</body>
</html>
```


서블릿에서 한글 요청 처리

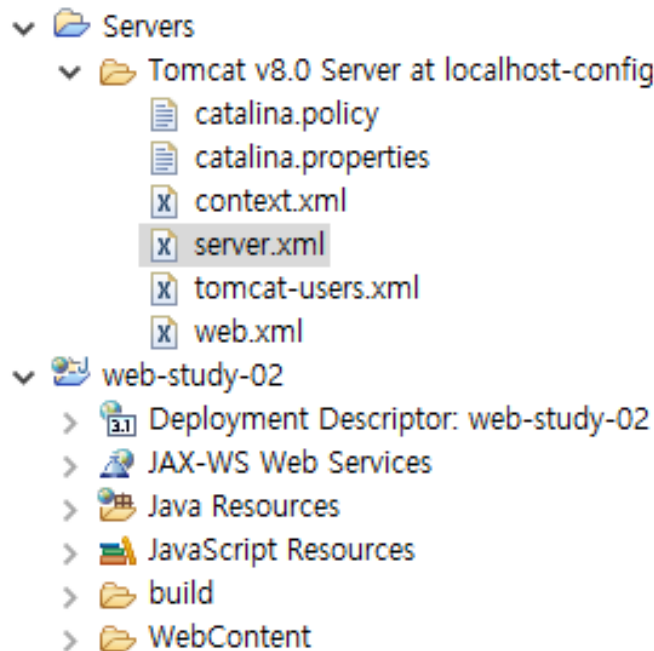
❖ InfoServlet.java

```
package unit06;
:
@WebServlet("/InfoServlet")
public class InfoServlet extends HttpServlet {
:
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String name = request.getParameter("name");
        String addr = request.getParameter("addr");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");
        out.println("당신이 입력한 정보입니다.<br>");
        out.println("이름 : ");
        out.println(name);
        out.println("<br> 주소 : ");
        out.println(addr);
        out.println("<br><a href='javascript:history.go(-1)'>다시</a>");
        out.print("</body></html>");
        out.close();
    }
}
```

서블릿에서 한글 요청 처리

❖ GET 방식의 한글 처리

- 요청 URL에 한글이 포함
- 톰캣 서버가 관장
 - URL 문자셋 디폴트는 운영체제의 문자셋을 사용(EUC-KR)
 - 톰캣의 문자셋을 UTF-8로 수정
 - 톰캣의 환경설정 파일인 `server.xml`에서 지정
 - `<Connector>`태그를 찾아 `URIEncoding` 속성을 UTF-8로 추가



서블릿에서 한글 요청 처리

❖ GET 방식의 한글 처리

- 톰캣의 server.xml 수정
 - <Connector>태그중 서비스 포트번호를 찾아 URLEncoder 속성 추가

```
<Connector connectionTimeout="20000"  
    port="8081"  
    protocol="HTTP/1.1"  
    URLEncoder="UTF-8"  
    redirectPort="8443"/>
```

- 서버 재기동하여 설정변경 반영

서블릿에서 한글 요청 처리

❖ POST 방식의 한글 요청

- 톰캣 설정만으로 안됨
- 06_info.jsp

```
<h3> get 방식에서 한글 깨짐 방지 </h3>
<form method="get" action="InfoServlet">
  이름 : <input type="text" name="name"><br>
  주소 : <input type="text" name="addr"><br>
  <input type="submit" value="전송" >
</form>
<br>
<h3> post 방식에서 한글 깨짐 방지 </h3>
<form method="post" action="InfoServlet">
  이름 : <input type="text" name="name"><br>
  주소 : <input type="text" name="addr"><br>
  <input type="submit" value="전송" >
</form>
```

서블릿에서 한글 요청 처리

❖ POST 방식의 한글 요청

- InfoServlet.java 수정

```
public class InfoServlet extends HttpServlet {  
    :  
  
    protected void doPost(HttpServletRequest request,  
                           HttpServletResponse response)  
        throws ServletException, IOException {  
        doGet(request, response);  
    }  
  
}
```

서블릿에서 한글 요청 처리

❖ POST 방식의 한글 요청

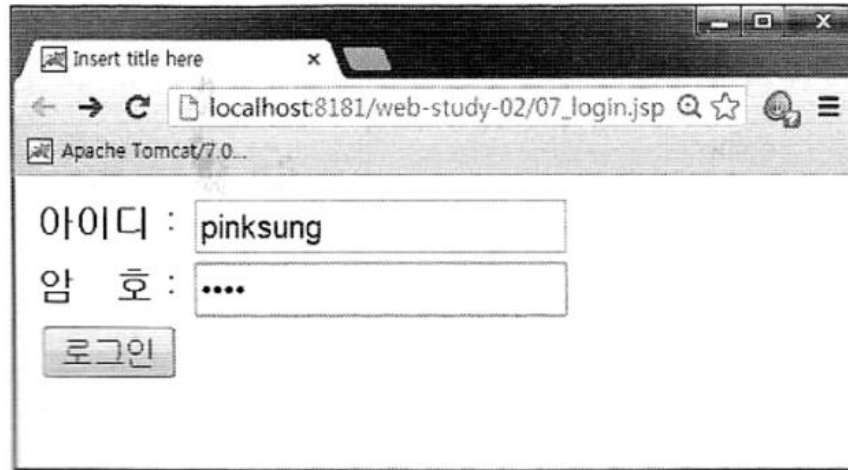
- POST 방식은 폼 데이터를 HTTP 프로토콜의 body 영역에 넣어 전송
- 톰캣이 관장하지 않음
 - 프로그램에서 처리해야 함.
 - request 객체의 setCharacterEncoding() 메서드로 문자셋 지정 필요
 - 이후에 요청 파라미터 추출

```
public class InfoServlet extends HttpServlet {  
    :  
    protected void doPost(HttpServletRequest request,  
                           HttpServletResponse response)  
        throws ServletException, IOException {  
  
        request.setCharacterEncoding("UTF-8");  
        doGet(request, response);  
    }  
  
}
```

기타 다양한 입력 양식

❖ 비밀번호 입력 상자

- type="password"



```
<input type="password" name="pwd">
```

기타 다양한 입력 양식

❖ label

- 입력 양식에 대한 설명
 - for 속성으로 연관된 입력요소 id 지정
 - label을 클릭하면 연관 입력요소 반응(라디오버튼, 체크박스에서 유용)

```
<label for="userid"> 아이디 : </label>  
<input type="text" name="id" id="userid"><br>
```


기타 다양한 입력 양식

❖ 로그인 폼 처리

파일 이름	설명
07_login.jsp	아이디와 암호를 입력 받는 폼 양식을 갖는다. [전송] 버튼을 클릭하면 서블릿으로 입력된 값이 전송되고 서블릿에서 입력받은 값을 받아 처리한다.
LoginServlet.java	HTML 문서의 입력 양식에 입력한 값을 처리하는 서블릿 클래스이다.

기타 다양한 입력 양식

❖ 07_login.jsp

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
    pageEncoding="UTF-8"%>
:
<body>
    <form method="post" action="LoginServlet">
        <label for="userid"> 아이디 : </label>
        <input type="text" name="id" id="userid"><br>
        <label for="userpwd"> 암호 : </label>
        <input type="password" name="pwd" id="userpwd"><br>
        <input type="submit" value="로그인">
    </form>
</body>
</html>
```

기타 다양한 입력 양식

❖ LoginServlet.java

```
package unit07;
:
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
    :
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String id = request.getParameter("id");
        String pwd = request.getParameter("pwd");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");
        out.println("당신이 입력한 정보입니다.<br>");
        out.println("아이디 : ");
        out.println(id);
        out.println("<br> 비밀번호 : ");
        out.println(pwd);
        out.println("<br><a href='javascript:history.go(-1)'>다시</a>");
        out.print("</body></html>");
        out.close();
    }
}
```

기타 다양한 입력 양식

❖ 로그인 폼 처리

- 비밀번호를 GET 방식으로 전송하는 경우 URL에 노출
- POST 방식으로 처리해야 함.

기타 다양한 입력 양식

❖ <textarea>

- 여러 줄 글자 입력

속성	설명
rows	텍스트 영역의 높이를 지정한다.
cols	텍스트 영역의 너비를 지정한다.
name	텍스트 영역이 여러 개일 경우 name 속성에 특정 이름을 지정하면 어느 입력 양식에 입력된 데이터인지를 구분할 수 있다.

```
<textarea rows="5" cols="80"></textarea>
```

기타 다양한 입력 양식

❖ radio 버튼

- type="radio"
- 여러 선택 요소 중 하나만 선택
 - 그룹 단위로 하나 선택
 - name이 같으면 동일 그룹

```
<input type="radio" name="그룹명">
```

- value 속성을 지정하여 해당 버튼이 선택시 전송할 값 지정

기타 다양한 입력 양식

❖ 예제

파일 이름	설명
08_radio.jsp	라디오 버튼과 <textarea> 태그가 있는 폼 양식을 갖는다. [전송] 버튼을 클릭하면 서블릿으로 선택된 값이 전송되고 서블릿에서 선택된 값을 받아 처리한다.
RadioServlet.java	HTML 문서에서 선택된 라디오 버튼과 <textarea> 태그 값을 처리하는 서블릿 클래스이다.

기타 다양한 입력 양식

❖ 08_radio.jsp

```
<form method="get" action="RadioServlet">
  <label for="gender"> 성별 : </label>
  <input type="radio" id="gender"
    name="gender" value="남자" checked> 남자
  <input type="radio" id="gender"
    name="gender" value="여자"> 여자 <br>
  <br>
  <label for="chk_mail"> 메일 정보 수신 여부 : </label>
  <input type="radio" id="chk_mail" name="chk_mail"
    value="yes" checked> 수신
  <input type="radio" id="chk_mail" name="chk_mail" value="no">
  거 부<br>
  <br>
  <label for="content"> 간단한 가입 인사를 적어주세요^o^ </label>
  <textarea id="content" name="content"
    rows="3" cols="35"></textarea><br>
  <input type="submit" value="전송">
</form>
```

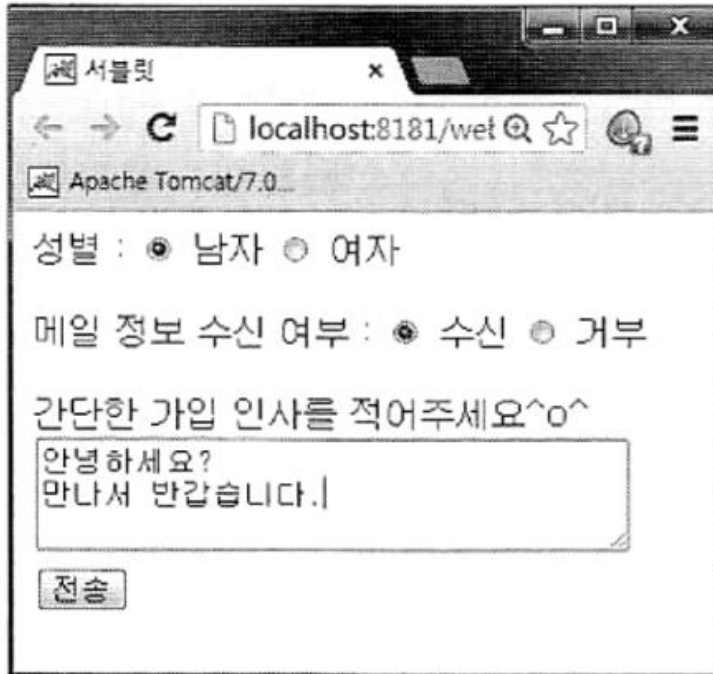

기타 다양한 입력 양식

❖ RadioServlet.java

```
protected void doGet(...) {
    response.setContentType("text/html;charset=UTF-8");
    String gender = request.getParameter("gender");
    String chk_mail = request.getParameter("chk_mail");
    String content = request.getParameter("content");
    PrintWriter out = response.getWriter();
    out.print("<html><body>");
    out.println("당신이 입력한 정보입니다.<hr>");
    out.println("성별 : <b>");
    out.println(gender);
    out.println("</b><br> 메일 정보 수신 여부 : <b>");
    out.println(chk_mail);
    out.println("</b><br> 가입 인사 : <b><pre>");
    out.println(content);
    out.println("</pre></b><a href='javascript:history.go(-1)''>다시</a>");
    out.print("</body></html>");
    out.close();
}
```

기타 다양한 입력 양식

❖ 예제



서블릿 x

localhost:8181/web-stud

Apache Tomcat/7.0...

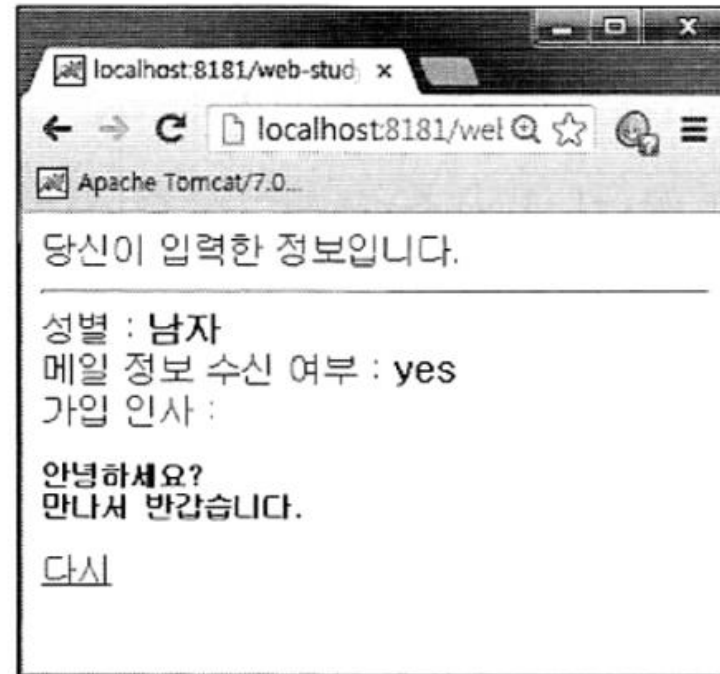
성별 : ☒ 남자 ☐ 여자

메일 정보 수신 여부 : ☒ 수신 ☐ 거부

간단한 가입 인사를 적어주세요^o^

안녕하세요?
만나서 반갑습니다.

전송



localhost:8181/web-stud x

localhost:8181/web-stud

Apache Tomcat/7.0...

당신이 입력한 정보입니다.

성별 : 남자

메일 정보 수신 여부 : yes

가입 인사 :

안녕하세요?
만나서 반갑습니다.

다시

기타 다양한 입력 양식

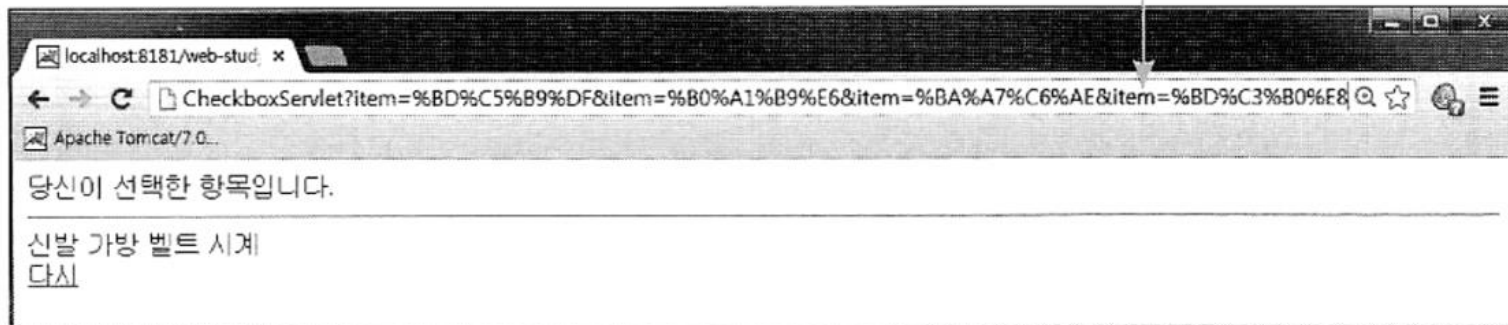
❖ 체크 박스

- type="checkbox"
- 다중 선택 항목 운영시 사용
- 같은 name으로 여러 요소가 운영
- 복수개의 값이 서버로 전달됨 → 서버에서 배열 처리 필요



체크 박스 4개를 선택했다. 체크 박스의 이름이 모두 item으로 되어 있기 때문에 쿼리 스트링에 선택된 항목에 대한 이름과 값이 전송될 때 item이란 동일한 이름으로 4번 넘겨진다.

```
String items[]=request.getParameterValues("item");
```



기타 다양한 입력 양식

❖ 체크 박스

- `request.getParameterValues()`를 이용하여 배열 추출

❖ 예제

파일 이름	설명
09_checkbox.jsp	체크 박스가 있는 폼 양식을 갖는다. [전송] 버튼을 클릭하면 서블릿으로 다중 선택된 값이 전송되고 서블릿에서 선택된 값을 받아 처리한다.
CheckboxServlet.java	HTML 문서에서 선택된 체크 박스 값을 처리하는 서블릿 클래스이다.

기타 다양한 입력 양식

❖ 09_checkbox.jsp

```
<h2>악세사리</h2>
관심항목을 선택하세요.
<hr>
<form method="get" action="CheckboxServlet">
  <input type="checkbox" name="item" value="신발"> 신발
  <input type="checkbox" name="item" value="가방"> 가방
  <input type="checkbox" name="item" value="벨트"> 벨트
  <br>
  <input type="checkbox" name="item" value="모자"> 모자
  <input type="checkbox" name="item" value="시계"> 시계
  <input type="checkbox" name="item" value="주얼리"> 주얼리
  <br>
  <input type="submit" value="전송">
</form>
```

기타 다양한 입력 양식

❖ CheckboxServlet.java

```
protected void doGet(...) {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.print("<html><body>");
    String items[] = request.getParameterValues("item");
    if (items == null) {
        out.print("선택한 항목이 없습니다.");
    } else {
        out.println("당신이 선택한 항목입니다.<hr>");
        for (String item : items) {
            out.print(item + " ");
        }
    }
    out.println("<br><a href='javascript:history.go(-1) '>다시</a>");
    out.print("</body></html>");
    out.close();
}
```

기타 다양한 입력 양식

❖ <select>

- 목록 상자
- 하위 항목을 <option>태그로 지정
- name 속성은 <select>태그에 지정, value 속성은 <option>태그에 지정
- <option>태그의 value 속성 생략시 <option>태그 텍스트가 값으로 지정

```
<label for="job">직업</label>
<select id="job" name="job" size="1">
  <option value="">선택하세요</option>
  <option value="학생">학생</option>
  <option value="컴퓨터/인터넷">컴퓨터/인터넷</option>
  <option value="언론">언론</option>
  <option value="공무원">공무원</option>
  <option value="군인">군인</option>
  <option value="서비스업">서비스업</option>
  <option value="교육">교육</option>
</select>
```



기타 다양한 입력 양식

❖ <select>

- 다중 선택 모드 가능
 - multiple 속성 지정
 - 체크박스처럼 배열처리 필요

```
<label for="interest" style="float: left;">관심분야</label>
<select
  id="interest" name="interest" size='5'
  multiple="multiple">
  <option value="에스프레소">에스프레소</option>
  <option value="로스팅">로스팅</option>
  <option value="생두">생두</option>
  <option value="원두">원두</option>
  <option value="핸드드립">핸드드립</option>
</select>
```



기타 다양한 입력 양식

❖ 예제

파일 이름	설명
10_select.jsp	콤보 박스와 리스트 박스로 원하는 항목을 선택하는 폼 양식을 갖는다.
SelectServlet.java	HTML 문서에서 선택된 항목을 얻어와 출력하는 서블릿 클래스이다.

기타 다양한 입력 양식

❖ 10_select.jsp

```
<form method="get" action="SelectServlet">
  <span style="float: left; margin-right: 20px">
    <label for="job">직업</label>
    <select id="job" name="job" size="1">
      <option value="">선택하세요</option>
      <option value="학생">학생</option>
      <option value="컴퓨터/인터넷">컴퓨터/인터넷</option>
      <option value="언론">언론</option>
      <option value="공무원">공무원</option>
      <option value="군인">군인</option>
      <option value="서비스업">서비스업</option>
      <option value="교육">교육</option>
    </select>
  </span>
```

기타 다양한 입력 양식

❖ 10_select.jsp

```
<label for="interest" style="float: left;">관심분야</label>
<select
  id="interest" name="interest" size='5'
  multiple="multiple">
  <option value="에스프레소">에스프레소</option>
  <option value="로스팅">로스팅</option>
  <option value="생두">생두</option>
  <option value="원두">원두</option>
  <option value="핸드드립">핸드드립</option>
</select>
<br><br>

<input type="submit" value="전송"
  style="float: right; margin-right: 50px">
</form>
```

기타 다양한 입력 양식

❖ SelectServlet.java

```
protected void doGet(...){
    String job = request.getParameter("job");
    String interests[] = request.getParameterValues("interest");
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.print("<html><body>");
    out.println("당신이 선택한 직업 : <b>");
    out.print(job);
    out.println("</b><hr>당신이 선택한 관심 분야 : <b>");
    if (interests == null) {
        out.print("선택한 항목이 없습니다.");
    } else {
        for (String interest : interests) {
            out.print(interest + " ");
        }
    }
    out.println("</b><br><a href='javascript:history.go(-1)''>다시</a>");
    out.print("</body></html>");
    out.close();
}
```

기타 다양한 입력 양식

❖ 예제

