

02. Servlet

1) Servlet (Server Side Applet)

: 서버상에서 실행되는 웹프로그램 ➔ 브라우저에서 실행결과를 볼 수 있다.
: Server + Let(허용)의 합성어
: 서버(Server)에서 애플리케이션(Application)을 허용(Let)한다 라는 의미

2) Servlet 의 구성

: 웹언어+자바프로그래밍 ➔ HTML,CSS,JS + Java

3) Servlet 실행 절차

- ① 클라이언트로부터 HelloServlet 요청
- ② 서블릿 컨테이너에서 해당 서블릿(HelloServlet) 객체 검색
(있다면 진행, 없다면 생성)
- ③ 해당 서블릿(HelloServlet)을 처리하기 위해 데이터베이스 작업이 필요하면
데이터베이스 연결 및 처리
- ④ 응답전송

4) Servlet 의 활용방식

웹프로그래밍을 할 때

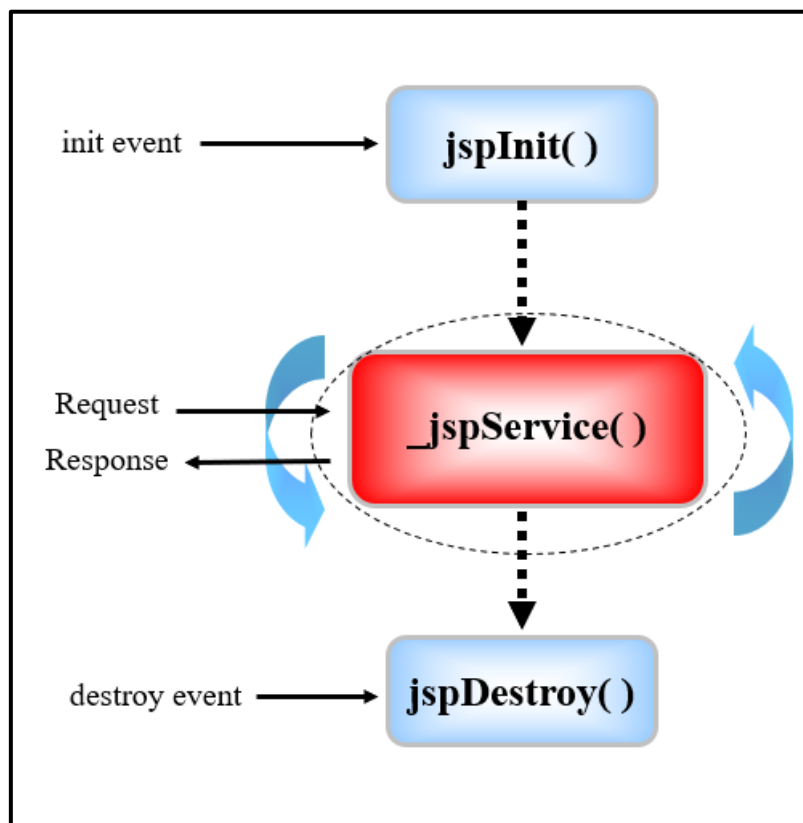
1. 서블릿을 먼저 작성하고 ➔ JSP 로 넘겨서 실행하는 방식
7~80% 자바코드 + 15~20% HTML, CSS
단점 : 접근이 어려움
장점 : jsp 보다 속도가 빠르다. (내부적으로 변환하는 시간이 없다)
2. 바로 JSP 로 작업 (MODEL 1)
7~80% HTML,CSS,JS + 15~20% 최소한의 자바코드
단점 : 자바코드 + 디자인코드가 혼재 ➔ 에러가 발생, 찾기 어렵다.

3. Servlet 과 JSP 를 혼합해서 사용 (MODEL2) ➔ 우리가 해야하는 것

5) Servlet 을 만들수 있는 조건

- ① `import javax.servlet.*;` // 서블릿의 클래스
`import javax.servlet.http.*;` // 웹상에서 접속에 관련된 정보 (프로토콜)
두 클래스를 import 시켜줘야한다.
→ tomcat8.5\lib\servlet-api.jar 파일에 저장되어 있음
- ② 반드시 `public class` 로 작성되어야 한다.
: 누구나 접속이 가능하게 만들기 위해
- ③ 반드시 `HttpServlet` 클래스를 상속받아야 한다.
get 방식으로 요청 → `doGet()` method 호출
post 방식으로 요청 → `doPost()` method 호출

서블릿 내부 작동 방식



HelloTest.java (Servlet)

```
import java.io.IOException; // 입출력 예외처리
import java.io.PrintWriter;
```

```
import javax.servlet.Servlet;
import javax.servlet.ServletConfig; // Configuration : Servlet 환경설정에 관련된 클래스
import javax.servlet.ServletException; // Exception : Servlet 오류 예외처리 클래스
import javax.servlet.annotation.WebServlet; // Annotation : Servlet 의 Annotation에 관련된 클래스
import javax.servlet.http.HttpServlet; // HttpServlet : Servlet을 만들기 위한 부모클래스 (상속해야한다)
import javax.servlet.http.HttpServletRequest; // request 역할 interface
import javax.servlet.http.HttpServletResponse; // response 역할 interface
```

1. import javax.servlet.*; // 서블릿
import javax.servlet.http.*; // 웹상 접속에 관련된 정보
2. 반드시 public class 로 작성해야 한다. : 누구나 접속할 수 있어야 한다.
3. 반드시 HttpServlet 클래스를 상속받아야된다. : doGet, doPost
: doGet > get으로 접근했을 때의 반응 -> doGet()을 자동으로 호출 (콜백메서드 계열)
: doPost > post로 접근했을 때의 반응 -> doPost() 을 자동으로 호출

```
/**
 * Servlet implementation class HelloTest
 */
@WebServlet("/HelloTest") // 어노테이션이 없으면, xml 환경설정을 불러올 수 없다.
```

```
public class HelloTest extends HttpServlet {
    private static final long serialVersionUID = 1L;
    /**
     * @see Servlet#init(ServletConfig)
     */
    public void init(ServletConfig config) throws ServletException {
        // 가장 먼저 실행됨
        // 생성자와 같은 역할(서블릿의 초기값을 설정할 때 필요)
    }

    /**
     * @see Servlet#destroy()
     */
    public void destroy() {
        // Servlet 종료시 실행되는 method → 메모리해제
    }
}
```

사용자의 요청(Get 방식)을 받을 때 자동으로 처리해주는 method
요청을 받아서 처리해주는 메서드 **(request,response){기능 구현}**

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    // TODO Auto-generated method stub
```

//callback Method(운영체제가 호출)

➔ 웹상에서 get방식으로 접속할때마다 자동으로 호출되는 메서드

//어떤형태의 문서를 보내줄지 결정

response.setContentType("text/html;charset=utf-8"); // html문서 + 한글데이터로 해주세요

//PrintWriter out=new PrintWriter(~);

PrintWriter out=response.getWriter();//객체 생성 3번째 방법



//1. 어떤형태의 문서를 만들어서 보내줄지 결정 -> **response** 객체가 처리
response.setContentType("text/html;charset=utf-8"); //html 문서로 처리 (한글)

//2. Print Writer 클래스의 method 반환형을 통해서 얻어오기

PrintWriter out = response.getWriter();

out.println("<html> <header> </header>"); // document.write("<html>");

out.println("<body>");

//외부에 보여줄 내용

//추가

out.println("<table border=1>");

for(int i=2; i<=9; i++) { //단

out.println("<tr>");

for(int j=1; j<=9; j++) {

out.println("<td>");

out.println(i+"*" +j+"="+ (j*i));

out.println("</td>");

}

out.println("</tr>");

}

out.println("</table>");

out.println("</body>");

out.println("</html>");

}

HTML 구현부분

**** 서블릿을 실행시킬때의 요청경로 ****

➔ `http://localhost:8080/ServletTest/HelloTest`

/프로젝트명/실행시킬 서블릿클래스명

➔ URL Mapping부분

➔ xml 파일 설정=>모델2 (환경설정할때만 필요)=>스프링

`http://localhost:8080/ServletTest/test/imsi/HelloTest`

➔ 가상경로를 지정해서 사용할 수도 있다.(보안을 위해서)

@WebServlet("어떻게 접근하면 서블릿을 실행시킬수가 있는 상대경로를 지정")

※ doGet() or doPost() or doService()

: form 태그 선언시의 **method** 방식에 따라 정해진다.

(1) doGet() ➔ method : get : 전달되는 값이 **노출**된다

(2) doPost() ➔ method : post : 전달되는 값이 **노출되지 않는다**.

(3) service() : 두 방식 모두 가능

※ 팝업창을 열어보자

ServletTest

➔ src ➔ Notice.java : Servlet (파일의 내용을 불러와서 창에 출력)

➔ WebContent ➔ index.jsp 에서 : 이벤트처리(onload)

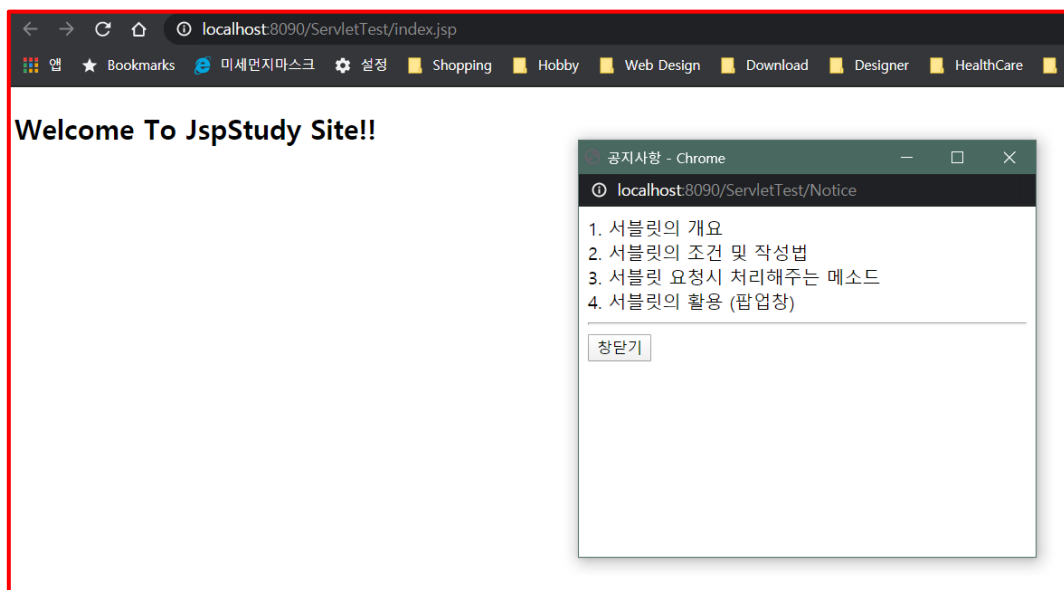
➔ main.jsp 에서 : 메인페이지(javascript)

> window.open() <-> window.close()

➔ notice -> 2020218.txt 파일을 입출력시킴

형식) `window.open(1.볼러올 파일명 2.문서의 타이틀제목 3.창의 옵션(width,height,left,top))`

```
<title>메인창을 실행</title>
<script>
function test(){
  //window.open('./sub/send.html','w','left=500,top=300,width=400,height=300') → send.html 오픈
  window.open('/ServletTest/Notice','w','left=300,top=200,width=400,height=300') → Notice Servlet
}
</script>
</head>
<body onload="test()">
  <h2>Welcome To JspStudy Site!!!</h2>
</body>
```



Notice.java (Servlet)

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

// TODO Auto-generated method stub

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

PrintWriter out=response.getWriter();

out.println("<html><head><title> 공지사항 </title></head>");

out.println("<body>");

//notice->2020218.txt

Calendar **cal=Calendar.getInstance()**;// Date d=new Date();

String fileName="";//불러올 파일명을 저장

➔ 파일명을 년도,월,일자 데이터를 읽어들이어서 합쳐서 만들 (String)

fileName+=cal.get(Calendar.YEAR);//2020

fileName+=cal.get(Calendar.MONTH)+1;//0~11+1=>1~12=>20202

fileName+=cal.get(Calendar.DATE);//18->윤년도 자동계산->2020218

fileName+=" .txt";//2020218.txt

//경로는 절대경로

String realPath="C:/webtest/4.jsp/sou/ServletTest/WebContent/notice/" + **fileName**;

System.out.println("realpath=>" + realPath);

try {

//FileInputStream(영문) or FileReader(한글)

BufferedReader br=new BufferedReader(new FileReader(realPath));

String line="";//한줄씩 읽어들이어서 저장할 변수

while((line=br.readLine())!=null) {//더이상 읽어들이 수 없을때까지

out.println(line+"
");//클라이언트의 브라우저로 전송

}

br.close();

}catch(IOException e) {

System.out.println("불러올 파일의 경로와 파일명을 확인하세요

=>" + e);//e.toString()

}catch(Exception e) {

System.out.println("오늘 공지사항이 없습니다." + e);

}

//-----

out.println("<p align=center>");

out.println("<hr>");

out.println("<input type='submit' value='창닫기' onclick='window.close()'> ");

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

}

java 코드

html 코드