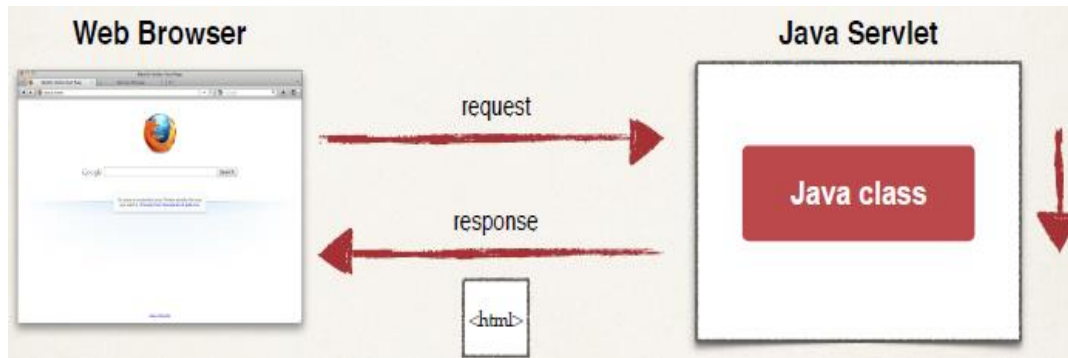


1. helloworld 서블릿 개요

서블릿이란?

- 웹서버상에서 처리되는 자바 클래스
- 자바 클래스가 웹브라우저에게 되돌려주는 HTML 생성
- HTML 폼 데이터를 읽을 수 있고, 쿠키와 세션을 사용할 수 있음
- 상위레벨에서 JSP와 유사한 기능



코드

`@WebServlet("/HelloWorldServlet")`
`public class HelloWorldServlet extends HttpServlet` HttpServlet에서 상속받아
오버라이드한 doGet() 메소드

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

Step1: 콘텐츠 타입(content type) 설정
`response.setContentType("text/html");`

Step2: 출력(PrintWriter) 설정
`PrintWriter out = response.getWriter();`

Step3: HTML 콘텐츠 생성
`out.println("<html><body>");`
`out.println("<h2>Hello World</h2>");`
`out.println("
");`
`out.println("Time on the server is: " + new java.util.Date());`
`out.println("</body></html>");`
}

Hello World

Time on the server is: Thu Dec 24 10:33:13 EST 2015

javaspservlet_study7
Loading descriptor for javaspservlet_study7..
JAX-WS Web Services
Java Resources
src
Libraries
JRE System Library [jdk-11.0.16.1]
Referenced Libraries
Server Runtime [Apache Tomcat v9.0]

servlet-api.jar - D:\apache-tomcat-9.0.40\lib\servlet-api.jar
javax.servlet
javax.servlet.annotation
javax.servlet.descriptor
javax.servlet.http
Cookie.class
CookieNameValidator.class
HttpFilter.class
HttpServlet.class

PrintWriter

- 바이트를 문자 형태를 가지는 객체로 변경
- 클라이언트에 문자형태의 응답을 위해 out 이라는 PrintWriter 클래스 객체를 정의
- PrintWriter 의 `getWriter()` 메소드를 통해 인스턴스 획득
- **out.print(HTML 태그)** 형태로 작성하여 스트림에 텍스트를 기록

```

package javax.servlet.http;

...

public abstract class HttpServlet extends GenericServlet {

...

    private static final String METHOD_GET = "GET";
    private static final String METHOD_POST = "POST";
    private static final String METHOD_PUT = "PUT";

...

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String msg = IStrings.getString("http.method_get_not_supported");
        sendMethodNotAllowed(req, resp, msg);
    }

...

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String msg = IStrings.getString("http.method_post_not_supported");
        sendMethodNotAllowed(req, resp, msg);
    }

...
}

```

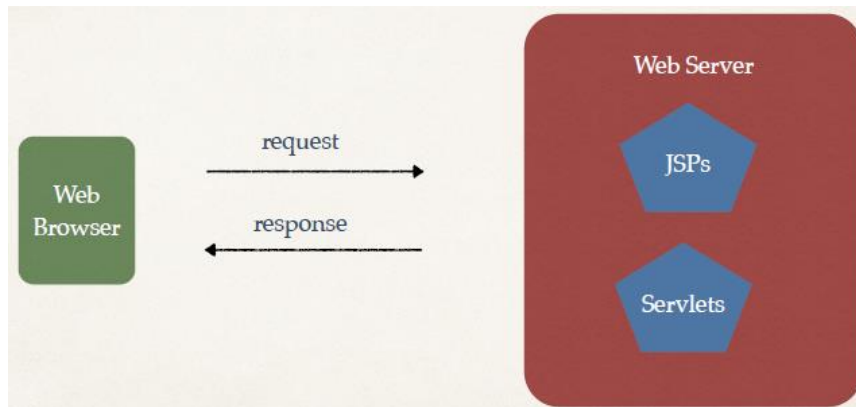
2. 서블릿 접근 : 애노테이션(WebServlet)내 지정된 경로(path) 사용)



실습1

실습2

3. JSP와 서블릿 비교



JSP와 서블릿 비교

| JSPs | Servlets |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">• HTML file with .jsp extension• Contains static HTML• JSP to generate HTML• Has built-in JSP objects | <ul style="list-style-type: none">• Java class file• Generate all HTML• More steps to access web objects |

어느 것을 사용해야 하나?

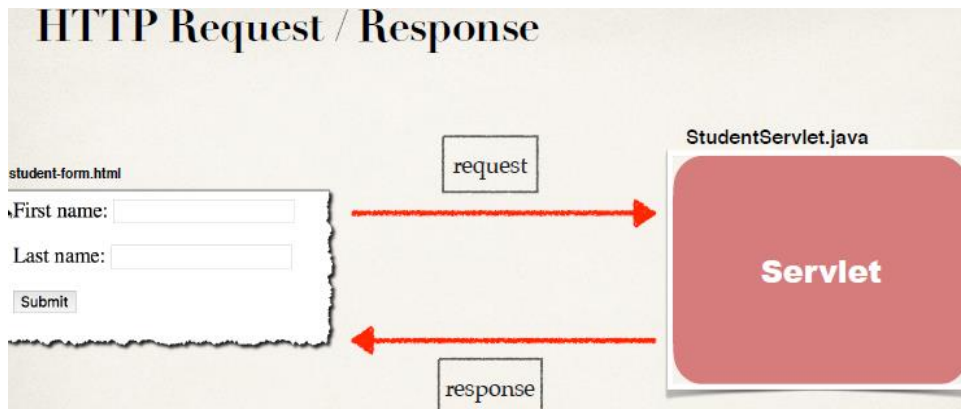
서블릿: 1997, JSP: 1999

- 자바 웹 애플리케이션 개발을 위해 어느 것을 사용할 수 있나?
- 사이트 전체를 서블릿으로만 빌드하기
- 사이트 전체를 JSP로만 빌드하기

베스트 프랙티스

- 둘을 통합!
- 서블릿: 비즈니스 로직
- JSP: 프리젠테이션 뷰(presentation view)를 핸들링(handle) the
- **MVC 디자인 패턴 적용**: Model-View-Controller (MVC)

4. 서블릿에서 폼 데이터 읽기



Step 1: Building HTML Form

```
<form action="StudentServlet" method="GET">  
  First name: <input type="text" name="firstName" />  
  Last name: <input type="text" name="lastName" />  
  <input type="submit" value="Submit" />  
</form>
```

A visual representation of the HTML form, showing 'First name:' and 'Last name:' text inputs and a 'Submit' button.

Form GET method calls Servlet `doGet()` method

```
<form action="StudentServlet" method="GET">  
  ...  
</form>
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
  ...  
}
```

The diagram shows the connection between the `GET` method in the HTML form and the `doGet()` method in the Servlet. A red circle highlights the `GET` attribute in the form tag, and another red circle highlights the `doGet` method name in the Servlet code. A red arrow points from the first circle to the second.

Step 2: Reading Form Data with Servlet

First name:

Last name:

```
request object는 HTML 폼 데이터를 포함하고 있음
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

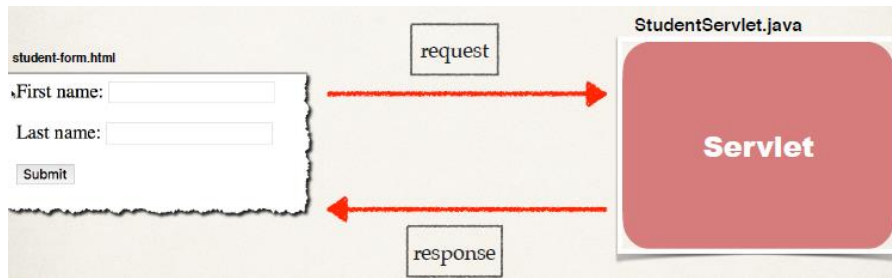
    ...
    out.println("The student is confirmed: " + request.getParameter("firstName")
        + " " + request.getParameter("lastName"));
}
```

실습

Step1: HTML 폼 만들기

Step2: 서블릿 만들기

5. 서블릿에서 GET과 POST의 차이



Form GET method calls Servlet `doGet()` method

```
<form action="StudentServlet" method="GET">
...
</form>
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    ...
    out.println(request.getParameter("firstName"));
}
```

Form POST method calls Servlet `doPost()` method

```
<form action="StudentServlet" method="POST">
...
</form>
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    ...
    out.println(request.getParameter("firstName"));
}
```

Sending Data with GET method

```
<form action="StudentServlet" method="GET">
...
</form>
```

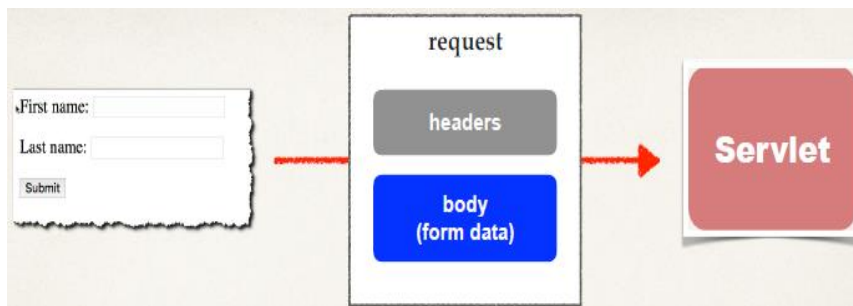
- Form data is added to end of URL as name/value pairs
- `theUrl?field1=value1&field2=value2...`

Sending Data with POST method

```
<form action="StudentServlet" method="POST">
...
</form>
```

- Form data is passed in the body of HTTP request message

브라우저 URL에서 볼 수 없다.



무엇을 사용해야 할까?



HTTP2.0인 현재는 글자수 제한 없음

6. 서블릿 파라미터 읽기

서블릿 설정 파라미터

- 웹 애플리케이션은 설정 파라미터를 사용할 수 있음
- 표준파일의 위치: [WEB-INF/web.xml](#)

Deployment Descriptor: web.xml

File: WEB-INF/web.xml

```
<web-app>
  <context-param>
    <param-name>max-shopping-cart-size</param-name>
    <param-value>99</param-value>
  </context-param>

  <context-param>
    <param-name>project-team-name</param-name>
    <param-value>The Coding Gurus</param-value>
  </context-param>
  ...
</web-app>
```

Reading Configuration Parameters

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

    ...
    ServletContext context = getServletContext();

    String maxCart = context.getInitParameter("max-shopping-cart-size");
    String teamName = context.getInitParameter("project-team-name");

    out.println("Max cart: " + maxCart);
    out.println("<br/><br/>");
    out.println("Team name: " + teamName);
}
```

```
<web-app>
  <context-param>
    <param-name>max-shopping-cart-size</param-name>
    <param-value>99</param-value>
  </context-param>

  <context-param>
    <param-name>project-team-name</param-name>
    <param-value>The Coding Gurus</param-value>
  </context-param>
</web-app>
```

실습

Step 1: web.xml에 설정 파라미터 정의

Step 2: 설정 파라미터를 읽기 위해 서블릿 만들기

← → ↻ ⓘ localhost:8080/javajspServlet_study7/TestParamServlet

Max cart: 99

Team name: 김형태와 친구들