TheWIZ

# 5강. Servlet 본격적으로 살펴보기-I

- 프로젝트 만들기
- doGet()
- doPost()
- 컨텍스트 패스(Context Path)

*Lecturer* Kim Myoung-Ho
*Nickname* 블스
blogstudy@naver.com

## 5-1. 프로젝트 만들기

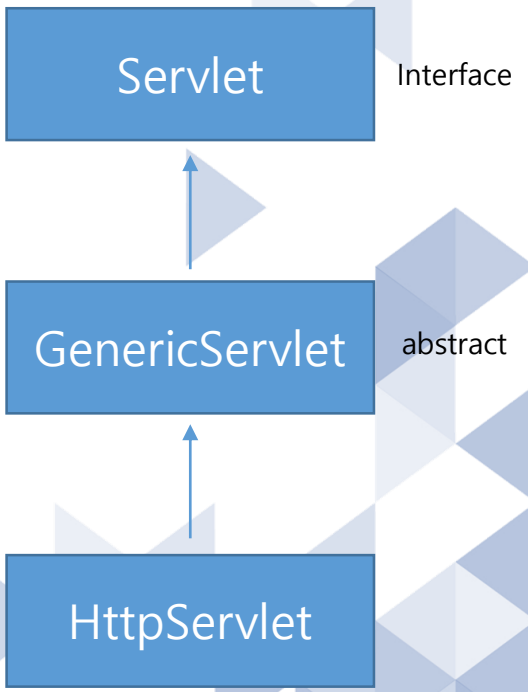
Servlet은 JAVA언어를 사용하여 웹프로그램을 제작하는 것 입니다. 간단한 Servlet 프로젝트를 만들어 보면서 전체적인 구조(흐름)를 살펴보도록 합니다.
(jsp_5_1_ex1_servletex)

- Servlet클래스는 HttpServlet 클래스를 상속 받음.

```
/**
 * Servlet implementation class HelloWorld
 */
@WebServlet("/HelloWorld")
public class HelloWorld extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
```

HttpServlet 클래스를 상속



## 5-1. 프로젝트 만들기

- 요청처리객체 및 응답처리객체를 톰캣에서 받음.

요청처리객체 / 응답처리객체

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    System.out.println("HelloWorld~~");

    response.setContentType("text/html");
    PrintWriter writer = response.getWriter();

    writer.println("<html>");
    writer.println("<head>");
    writer.println("</head>");
    writer.println("<body>");
    writer.println("<h1>HelloWorld~~~~</h1>");
    writer.println("</body>");
    writer.println("</html>");

    writer.close();

}
```

Console 출력

응답을 해줄때는 html 파일 형식으로 응답한다.

웹브라우저에 출력하기 위한 스트림

html

## 5-1. 프로젝트 만들기

- GET & POST 방식

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    System.out.println("HelloWorld~~");

    response.setContentType("text/html");
    PrintWriter writer = response.getWriter();

    writer.println("<html>");
    writer.println("<head>");
    writer.println("</head>");
    writer.println("<body>");
    writer.println("<h1>HelloWorld~~~</h1>");
    writer.println("</body>");
    writer.println("</html>");

    writer.close();

}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}
```

doGet() 호출

doPost() 호출

**Form태그 method 속성값 = get**

**GET 방식 :**
URL값으로 정보가 전송되어 보안에 약함.

**POST 방식 :**
header를 이용해 정보가 전송되어 보안에 강함.

**Form태그 method 속성값 = post**

## 5-2. doGet()

- html내 form태그의 method속성이 get일 경우 호출 됩니다.
- 웹브라우저의 주소창을 이용하여 servlet을 요청한 경우에도 호출 됩니다.

doGet메소드는 매개변수로 HttpServletRequest와 HttpServletResponse를 받습니다

## 5-2. doGet()

HttpServletResponse 객체의 setContentType() 메소드 호출하여 응답방식 결정 합니다.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    System.out.println("HelloWorld~~");

    response.setContentType("text/html; charset=euc-kr");
    PrintWriter writer = response.getWriter();

    writer.println("<html>");
    writer.println("<head>");
```

HttpServletResponse 객체의 getWriter() 메소드를 이용하여 출력 스트림을 얻습니다.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    System.out.println("HelloWorld~~");

    response.setContentType("text/html; charset=euc-kr");
    PrintWriter writer = response.getWriter();

    writer.println("<html>");
    writer.println("<head>");
```

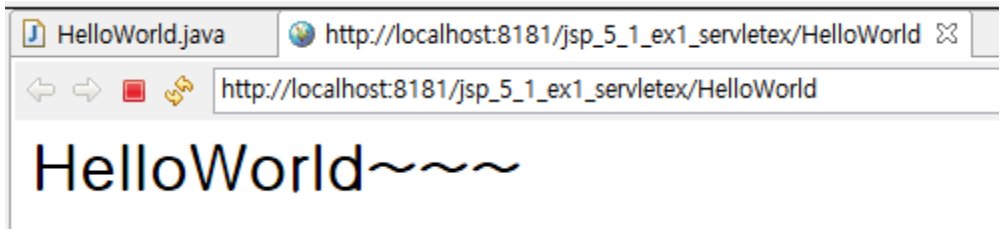
## 5-2. doGet()

출력스트림의 println() 메소드를 이용하여 출력하면, 웹브라우저에 출력 됩니다.

```
response.setContentType("text/html; charset=euc-kr");
PrintWriter writer = response.getWriter();

writer.println("<html>");
writer.println("<head>");
writer.println("</head>");
writer.println("<body>");
writer.println("<h1>HelloWorld~~~</h1>");
writer.println("</body>");
writer.println("</html>");

writer.close();
```




마지막에 출력객체 닫습니다.

```
writer.close();
```

## 5-3. doPost()

- html내 form태그의 method속성이 post일 경우 호출 됩니다.
(jsp_5_1_ex1_servletex)

post.html | PostMethod.java | http://localhost:8181/jsp_5_1_ex1_servletex/PostMethod

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>

    <form action="PostMethod" method="post">
        <input type="submit" value="post">
    </form>

</body>
</html>
```

**HTML**

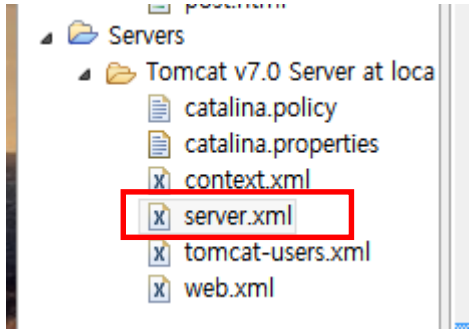
```java
protected void doPost(HttpServletRequest request, HttpServletResponse response) throw
    // TODO Auto-generated method stub
    System.out.println("doPost");

    response.setContentType("text/html; charset=euc-kr");
    PrintWriter writer = response.getWriter();
    writer.println("<html>");
    writer.println("<head>");
    writer.println("</head>");
    writer.println("<body>");
    writer.println("<h1>POST 방식 입니다. 따라서 getPost 메소드 호출 되었습니다.</h1>");
    writer.println("</body>");
    writer.println("</html>");
}
```

**Servlet**

## 5-4. 컨텍스트 패스(Context Path)

WAS(Web Application Server)에서 웹어플리케이션을 구분하기 위한 path 입니다.
이클립스에서 프로젝트를 생성하면, 자동으로 server.xml에 추가 됩니다.




post.html | PostMethod.java | http://localhost:8181/jsp_5_1_ex1_servletex/PostMethod | server.xml

```
101         /docs/config/cluster.html (reference documentation) -->
102       <!--
103       <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
104       -->
105
106       <!-- Use the LockOutRealm to prevent attempts to guess user passwords
107            via a brute-force attack -->
108       <Realm className="org.apache.catalina.realm.LockOutRealm">
109         <!-- This Realm uses the UserDatabase configured in the global JNDI
110              resources under the key "UserDatabase".  Any edits
111              that are performed against this UserDatabase are immediately
112              available for use by the Realm.  -->
113         <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
114       </Realm>
115
116       <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">
117
118         <!-- SingleSignOn valve, share authentication between web applications
119              Documentation at: /docs/config/valve.html -->
120         <!--
121         <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
122         -->
123
124         <!-- Access log processes all example.
125              Documentation at: /docs/config/valve.html
126              Note: The pattern used is equivalent to using pattern="common" -->
127         <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="%h %l %u %t &quot;%r&quot; %s %b" prefix="localhost_access_log." suffix=".txt"/>
128
129       <Context docBase="jsp_5_1_ex1_servletex" path="/jsp_5_1_ex1_servletex" reloadable="true" source="org.eclipse.jst.jee.server:jsp_5_1_ex1_servletex"/></Host>
130     </Engine>
131   </Service>
```