

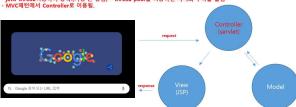
2장. Servlet– I



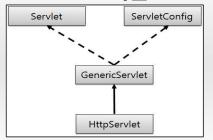
1. Servlet이란?

Servlet

- Servlet은 JSP표준이 나오기 전에 만들어진 표준으로 자바로 웹어플리케이션을 개발할 수 있도록 하기 위해 만들어짐.
- 서블릿을 이용하면 자바 클래스를 이용해서 웹어플리케이션을 개발할 수 있다. 반드시 매핑을 해야 함.(경로명 및 보안상의 문제 뒤에서 더 설명함)
- 동적 웹어플리케이션 컴포넌트.(jsp와 동일)
- .java 확장자.
- 클라이언트의 요청에 동적으로 작동하고, 응답은 html을 이용. java thread이용하여 동작.(가장 큰 장점) thread pool을 이용하면 서버의 부하를 줄임







GenericServlet 추상클래스는 Servlet과 ServletConfig 인터페이스를 구현함 HttpServlet은 GenericServlet 추상글래스를 상속받음





1. 프로젝트 생성





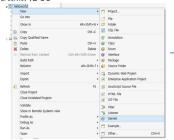
1. 프로젝트 생성







2. servlet파일 생성





서블릿 파일은 HttpServlet파일을 상속받아야 비로서 서블릿 파일이 된다는 것에 주목하자.



2. servlet파일 생성





서불럿 파일을 만들고 나서, URL 매명을 해주어야 하는데 두 가지 방법이 있다. 하나는 직접 위와 같이 해주고 @WebServite에 적용될 문자열을 입력하여 주는 방법 하나와, 두 번째 방법은 Web.xml파일에서 테그로 직접 지정해주는 방법이 있다. 매평은 그냥 일종의 닉데임이라고 생각하면 편할 것이다.



2. servlet파일 생성

9		Create Servlet	**	- 0
reate Serviet Enter serviet de	oloyment descrip	otor specific information.		(3)
Name: H Description:	elloWorld			
nitialization par	(0)	URL Mappings	×	
Name	Patterr /HW			Add
	-	OK	Cancel	Edit
		UK.	Cancel	Remove
/HelioWorld				Add.
/Heioricis				Edit
				Remove
Asynchronou	Support			
_				
?	< Back	Next >	Finish	Cancel



서블릿 파일을 만들고 나서, URL 매명을 해주어야 하는데 두 가지 방법이 있다. 하나는 직접 위와 같이 해주고 @WebServlet에 작용될 문자열을 입력하여 주는 방법 하나와, 두 번째 방법은 web.xmm파일에서 태그로 직접 지정해주는 방법이 있다. 매평은 그냥 일종의 닙네임이라고 생각하면 짜함 것이다.



2. servlet파일 생성











3. Servlet 실행

- Markers 🖽 Properties 👫 Servers 🙀 Data Source Explorer 🖺 Snippets 🕒 Console 🗵
- 고등 30, 2014 4:13:55 요후 org.apache.catalna.startup.HostConfig deployDirectory 정보: Deployment of web application directory C Navalec@pache-tomcat-7.0.57#apache-tomcat
- 12書 30, 2014 4:13:55 总基 org apache catalina startup HostConfig deployOrectory 製版: Deploying web application directory Citiavalectipache formati-7.0.57%apache-tomost-7.0.
- 12書 30, 2014 4:13:55 至車 org apache catalina startup.HostConfig deployDirectory
- 意識: Deployment of web application directory Chijavalechapache-tomcat-7.0.57wapache-tomca 12書 30. 2014 4:13:55 皇本 org apache covete AbstractProtocol start
- 정보: Starting ProtocolHandler ["http-bio-8181"]
- 警 30, 2014 4:13:55 全事 org apache coyote AbstractProtocol start 展: Starting Protocol Handler Calo-bio-8009*1
- 12書 30, 2014 4:18:55 卒革 org apache catalina startup.Catalina start 發展: Server startup in 1529 ms



3. web.xml에서 서블릿 매핑하기

너무 URL주소가 길고, 보안에 노출되어 있는 경로를 간단하게 맵핑하는 것이다.

기존 경로: http://localhost:8090 ₩Practice_Chap02₩src\sec01_exam_servlet\HelloWorld

URL맵핑 경로: http://localhost:8090/Practice Chap02/HWorld



3. web.xml에서 서블릿 매핑하기

```
Project Explorer 33
                                    HelioWorld lava R web.xml 22
                                       1 <?xml version="1,0" encoding="UTF-8">
 . W halloworld
                                      28 sweb-app xmlnsxsi="http://www.w3.org/2001/XMI Schema-in
   5 % Deployment Descriptor: helloworld
                                           <display-name>helloworld</display-name>
   A JAX-WS Web Services
                                      49 <welcome-file-list>
   . Pill Java Resources
                                             swelcome-files index html s/welcome-files
     4 (00 sec
                                            <welcome-file>index.htm</welcome-file>
        # III com lavalec.ex
                                            <welcome-file>index.jsp</welcome-file>
          → J) HelloWorld.java
                                            <welcome-file>default.html</welcome-file>
     n mà Libraries
   JavaScript Resources
                                            <welcome-file>default.htm</welcome-file>
                                            <welcome-file>default.isp</welcome-file>

■ WebContent

                                           </welcome-file-list>
     > META-INF
      WER-INF
                                            sservlet-name> helloworld s/servlet-name>
                                            <servlet-class>com.iavalec.ex.HelloWorld</servlet-class>
                                           </servlet>
                                           <servlet-mapping>
                                            <servlet-name>helloworld</servlet-name>
                                            <url-pattern>/hw</url-pattern>
                                           </servlet-mapping>
```

web.xml http://localhost:8181/helloworld/hw

http://localhost:8181/helloworld/hw

HelloWorld iava

<servlet-name> : 임의의 이름을 만들어 준다.

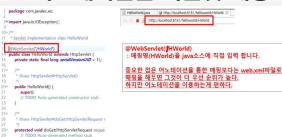
<servlet-class> : 매핑할 클래스 파일명을 패키지명을 포함하여 정확하게 입력한다.

<url-pattern>

: servlet-class의 클래스를 매핑할 임의 의 이름을 입력한다.주의할 점은 '/'로 시작해야 합니다.



4. 어노테이션을 서블릿 매핑하기



System.outprintln("HelloWorld~~"):



5. 서블릿 프로젝트 만들기

Serviet은 JAVA인어를 사용하여 웹 프로그램을 제작하는 것 입니다. 간단한 Serviet 프로젝트를 만들어 보면서 전체적인 구조(흐름)를 살펴보도록 합니다.

- Servlet클래스는 HttpServlet 클래스를 상속 받음.





5. 서블릿 프로젝트 만들기

	1	응답처리객체
isole 출력 🕳	protected void doGet(HttpServletRequest request, System.outprintln("HelloWorld~~");	HttpServletResponse response) throws ServletException, IOException {
	response.setContentType("text/html"); PrintWriter writer = response.getWriter();	
	writer.println(" <html>"); writer.println("<head>"); writer.println("</head>"); writer.println("<hody>"); writer.println("<hl>">+hl=loWorld~~~</hl>");</hody></html>	에 가장 등 이 에서 html = = 에 나 무이는 다. 13만 기는데 = # num= 일이기 때문에 서불맛과는 조금 다르다. 기억하자.)
	writer.println(""); writer.println("");	Client WAS DB
	writer.close();	request (IOIIICA) 클라이언트의 요청이 들어오면 서버에서 그것을 받아서

정보를 response객체에 담아서 넘겨준다.



5. 서블릿 프로젝트 만들기

- GET & POST 방식





6. doGet()

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

doGet메소드는 매개변수로 HttpServletRequest와 HttpServletResponse를 받습니다.(톰캣 서버가 자동 생성해 줌)

웹브라우저

doGet()

HttpServletRequest > 클라이언트의 요청 처리 객체

HttpServletResponse > 클라이언트에게 응답 처리 객체



6. doGet()

- 1. HttpServletResponse 객체의 setContentType() 메소드 호출하여 응답방식 결정합니다.
- 2. HttpServletResponse 객체의 getWriter() 메소드를 이용하여 출력 스트림을 얻습니다.

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, (DException {
System.outprinth("HelloWorld--");
기본적으로 응답할 때 html코드로 응답하고, 문자셋은 utf-8로 설정함(필히 설정해야 함)
response.setConnentType("text/html; charset= utf-8;
PrintWriter writer = response.getWriter();

writer.println("<html>");



6. doGet()

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

```
response.setContentType("text/fitml; charset=euc-ko");
PrintWriter writer = response.getWriter();
writer.print(n"charset);
```

마지막에 출력객체 닫습니다.(리소스 해제)

writer.close():



7. doPost()

- html내 form태그의 method속성이 post일 경우 호출 됩니다.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throw
post.html III PostMethod.java @ http://focalhost8181/jsp_5_1_ex1_servletex/PostMethod
                                                                                       // TODE Auto-generated method stud
  1 <IDOCTVPF html>
  20 shtml>
                                                                                       System.out.println("doPost"):
 30 cheads
  4 smeta charset=" utf-8 5
                                                                                      response.setContentType("text/html: charset=euc-kr");
  5 <title>Insert title here</title>
                                                                                      PrintWriter writer = response.getWriter();
  6 </head>
                                                                                      writer.println("<html>");
  78 <body>
                                                                                      writer.println("<head>");
                                                                                      writer.println("</head>"):
       <form action="PostMethod" method="post">
                                                                                      writer.println("<body>");
        <input type="submit" value="post">
                                                                                      writer.println("<h1>POST 방식 입니다. 따라서 getPost 메소드 호출 되었습니다.</h1>");
       </form>
                                                                                      writer.println("</body>");
                                                                                      writer.println("</html>"):
 13 </body>
 14 «/html»
```

HTML Servlet



8. 컨텍스트 패스(Context Path)

WAS(Web Application Server)에서 웹어플리케이션을 구분하기 위한 path입니다. (통상 웹어플리케이션(프로젝트명) 이름으로 봐도 무방하다.)

이클립스에서 프로젝트를 생성하면, 자동으로 server.xml에 추가 됩니다.

```
A Co Servers
   ♠ ○ Tomcat v7.0 Server at loca
         catalina policy
         atalina properties
         x context xml
         web.xml
```

```
«Realm className» "om anache catalina sealm (ocaCorillealm").
         «Realm className» "org.apache.catalina.realm.UserDatabaseRealm" resourceName» "UserDatabase") »
   «Host appBase» "avehapps" autoDeploy» "true" name» "localhost" uspackWARs» "true">
         «Valve className» "org.apache.catalina.valves.Accessl.ogValve" directory="logs" pattern="Wh M/ No Nr. Signot:Nr. Signo
         Contest docBase="list 5 | ext. servictes" path="list 5 | ext. servictes" reloadable="true" source="ora.eclipse.ist.ice.servenise 5 | ext. servictes") - ritiost;
```



감사합니다.