

```
1 1. Servlet 환경설정
2   1)Tomcat 9 설치
3   2)Apache 2.4.1과 Tomcat 9 연동하기
4
5
6 2. Tomcat Plug-in 설치
7   www.eclipse totale.com/tomcatPlugin.html
8   1)Eclipse > Help > Eclipse Marketplace
9   2)Find : TomcatPlugin
10  3)Eclipse Tomcat Plugin 9.1.4 > Install
11  4)After Installation, 관리자권한으로 Eclipse Start
12  5)Window > Preferences > Tomcat
13     -Tomcat version : Version 9.x
14     -Tomcat home : C:\Program Files\Apache Software Foundation\Tomcat 9.0
15     -server.xml
16  6)Tomcat > Advanced
17     -Tomcat base : C:\Program Files\Apache Software Foundation\Tomcat 9.0
18  7)Tomcat > Tomcat Manager
19     -ManagerApp username : admin
20     -ManagerApp password : javatocat
21  8)Apply and Close
22  9)반드시 OS의 Apache Tomcat Service 중지, 사용안함으로 설정
23  9)Tomcat Start : 문제 없음.
24  10)Tomcat Stop : port 가 정해져 있지 않아서 stop 없다는 에러 발생, 작업관리자에 계속 javaw.exe 있음.
25     -해결점 : server.xml의 다음을 수정
26     -변경전 : <Server port="-1" shutdown="SHUTDOWN">
27     -변경후 : <Server port="8005" shutdown="SHUTDOWN">
28
29
30 3. HelloServlet.java
31   import java.io.IOException;
32   import java.io.PrintWriter;
33   import javax.servlet.http.HttpServlet;
34   import javax.servlet.http.HttpServletRequest;
35   import javax.servlet.http.HttpServletResponse;
36   import javax.servlet.ServletException;
37
38   public class HelloServlet extends HttpServlet{
39       public void doGet(HttpServletRequest req,
40           HttpServletResponse res) throws IOException,ServletException{
41           PrintWriter out = res.getWriter();
42           out.println("<font size='7' color='blue'>Hello</font>");
43           out.close();
44       }
45   }
46
47
48 4. Servlet이란?
49   Server + Applet --> Server에서 실행되는 작은 Web Application
50   Servlet class가 Instance화 돼서 Memory에 올라온 객체
51
52
53 5. Servlet Project 시작하기
54   1)Tomcat의 docBase에 실제 Project_name의 물리적 폴더 생성
55
56   2)해당 프로젝트 이름의 Context 생성
57     -%CATALINA_HOME%/conf/server.xml
58     <Host..
```

```

59     <Context path="/Project_name" docBase="Physical Path in Explorer" />
60     </Host>
61     -ex
62     <Context path="/0316" docBase="C:/WebHome/0316" />
63
64     3)Project_name/WEB-INF/web.xml
65     <display-name>Welcome to Project_name</display-name>
66
67     4)Tomcat Manager에서 확인할 것
68
69
70 6. Servlet Service 등록하는 방법
71 1) 첫번째 방법 - annotation 을 이용 --> Servlet 3.0 부터 시작
72     -XML file 등을 이용하여 환경 설정 및 추가 정보를 등록하는 방법 대신에 Java Class code에 직접 설정하는 기술
73     로서 '@'로 시작된다.
74     @WebServlet("/servlet/Hello")
75     public class HelloServlet extends HttpServlet{
76
77 2) 두번째 방법 - web.xml 이용하기
78     -Annotation을 사용하지 않으려면 web.xml 파일의 <web-app>태그에 metadata-complete="true" 를
79     지정하면 된다.
80     <?xml version="1.0" encoding="ISO-8859-1"?>
81     <web-app xmlns="http://java.sun.com/xml/ns/javaee"
82     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
83     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
84     http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"
85     version="3.0"
86     metadata-complete="true">
87
88     <display-name>Welcome to My Servlet</display-name>
89     <description>
90     Welcome to Tomcat
91     </description>
92
93     <servlet>
94     <servlet-class>HelloServlet</servlet-class>
95     <servlet-name>Hello</servlet-name>
96     </servlet>
97     <servlet-mapping>
98     <servlet-name>Hello</servlet-name>
99     <url-pattern>/servlet/Hello</url-pattern>
100    </servlet-mapping>
101    </web-app>
102
103 -----
104
105 [Servlet 3.0 에 추가된 어노테이션]
106 @WebServlet           @Resources
107 @WebFilter            @PersistenceContext
108 @WebInitParam         @PersistenceContexts
109 @WebListener          @PersistenceUnit
110 @MultipartConfig      @PersistenceUnits
111 @DeclareRoles         @PostConstruct
112 @EJB                  @PreDestroy
113 @EJBs                 @RunAs
114 @Resource

```

```
115
116
117 7. @WebServlet 어노테이션 이용방법
118 1)Servlet WebMapping만 지정하는 방식
119 -web.xml
120 <?xml version="1.0" encoding="UTF-8"?>
121 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
122 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
123 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
124 http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"
125 version="3.1"
126 metadata-complete="false">
127
128 -@WebServlet\("/Mapping명"\)
129
130 2)추가 속성을 이용하는 방법
131 @WebServlet(name = "서블릿별명", urlPatterns = { "맵핑명", "/맵핑명2"})
132 @WebServlet(name="서블릿별명", value = { "/맵핑명", "/맵핑명2"})
133
134 ex) @WebServlet(name="MyServlet", urlPatterns={ "/xxx", "/yyy" })
135 --> http://localhost/ContextName/xxx or
136 --> http://localhost/ContextName/yyy 모두 가능
137
138
139 8. 한글 테스트
140 HanguelServlet.java
141
142 import java.io.IOException;
143 import java.io.PrintWriter;
144 import javax.servlet.http.HttpServlet;
145 import javax.servlet.http.HttpServletRequest;
146 import javax.servlet.http.HttpServletResponse;
147 import javax.servlet.ServletException;
148
149 public class HanguelServlet extends HttpServlet{
150     public void doGet(HttpServletRequest req,
151         HttpServletResponse res) throws IOException,ServletException{
152
153         res.setContentType("text/html;charset=UTF-8");
154         PrintWriter out = res.getWriter();
155         out.println("<meta charset='utf-8'>");
156         out.println("안녕하세요");
157         out.close();
158     }
159 }
160
161 <servlet>
162     <servlet-class>HanguelServlet</servlet-class>
163     <servlet-name>Hanguel</servlet-name>
164 </servlet>
165 <servlet-mapping>
166     <servlet-name>Hanguel</servlet-name>
167     <url-pattern>/servlet/Hanguel</url-pattern>
168 </servlet-mapping>
169
170
171 9. FirstServlet.java
172
```

[illegible]

```
231         <option value="서울">서울</option>
232         <option value="부산">부산</option>
233         <option value="대전">대전</option>
234         <option value="대구">대구</option>
235         <option value="인천">인천</option>
236         <option value="광주">광주</option>
237     </select>
238 </li>
239 <li><button>전송</button></li>
240 </ul>
241 </form>
242 </body>
243 </html>
```

245 2) GetPost.java

```
246 import java.io.FileWriter;
247 import java.io.IOException;
248 import java.io.PrintWriter;
249
250 import javax.servlet.ServletException;
251 import javax.servlet.annotation.WebServlet;
252 import javax.servlet.http.HttpServlet;
253 import javax.servlet.http.HttpServletRequest;
254 import javax.servlet.http.HttpServletResponse;
255
256 @WebServlet("/servlet/GetPost")
257 public class GetPost extends HttpServlet {
258     protected void doGet(HttpServletRequest request, HttpServletResponse response)
259         throws ServletException, IOException {
260         request.setCharacterEncoding("utf-8");
261         String username = request.getParameter("username");
262         String gender = request.getParameter("gender");
263         String hobbies = "";
264         String [] array = request.getParameterValues("hobby");
265         for(String str : array) hobbies += str + ",";
266         hobbies = hobbies.substring(0, hobbies.length() - 1);
267         int age = Integer.parseInt(request.getParameter("userage"));
268         String city = request.getParameter("usercity");
269
270         response.setContentType("text/html;charset=utf-8");
271         PrintWriter out = response.getWriter();
272         out.println("<h2>회원 정보</h2>");
273         out.println("<ul>");
274         out.println("<li>이름 : " + username + "</li>");
275         out.println("<li>성별 : " + gender + "</li>");
276         out.println("<li>취미 : " + hobbies + "</li>");
277         out.println("<li>나이 : " + age + "</li>");
278         out.println("<li>거주지 : " + city + "</li>");
279         out.println("</ul>");
280         out.close();
281     }
282     protected void doPost(HttpServletRequest request, HttpServletResponse response)
283         throws ServletException, IOException {
284         // TODO Auto-generated method stub
285         this.doGet(request, response);
286     }
287 }
```

```
288 3) 한글처리
289 a. POST 방식 --> req.setCharacterEncoding("utf-8");
290 b. GET 방식 : 수정 없음.
291
292 4) Telnet 을 이용한 http protocol 확인하기
293 Windows7 인 경우 telnet client 설치후 수행
294 telnet
295 set localecho
296 open localhost 8080
297 GET /getpost.html HTTP/1.1
298
299
300 11. Servlet의 초기화 Parameter 등록하기
301 1)web.xml이용하기
302
303 -InitParamServlet.java
304 package com.example;
305 import javax.servlet.ServletException;
306 import javax.servlet.http.HttpServlet;
307
308 public class InitParamServlet extends HttpServlet {
309     private String fileName;
310
311     @Override
312     public void init() throws ServletException {
313         fileName = getInitParameter("file");
314         System.out.println("파일명 : " + fileName);
315     }
316 }
317
318 -WEB-INF\web.xml
319 <servlet>
320     <servlet-name>InitParam</servlet-name>
321     <servlet-class>com.example.InitParamServlet</servlet-class>
322     <init-param>
323         <param-name>file</param-name>
324         <param-value>/WEB-INF/count.dat</param-value>
325     </init-param>
326 </servlet>
327 <servlet-mapping>
328     <servlet-name>InitParam</servlet-name>
329     <url-pattern>/servlet/InitParam</url-pattern>
330 </servlet-mapping>
331
332 -Console에서 결과 확인할 것
333 파일명 : /WEB-INF/count.dat
334
335 2)@WebInitParam 어노테이션을 이용한 초기화 파라미터 등록
336 -Servlet Code에서 @WebInitParam을 사용하여 초기화 Parameter 등록 가능
337 -InitParamAnnoServlet.java
338
339 package com.example;
340 import java.io.IOException;
341
342 import javax.servlet.ServletException;
343 import javax.servlet.annotation.WebInitParam;
344 import javax.servlet.annotation.WebServlet;
345 import javax.servlet.http.HttpServlet;
```

```
346 import javax.servlet.http.HttpServletRequest;
347 import javax.servlet.http.HttpServletResponse;
348
349 @WebServlet( name ="InitParamAnnoServlet" , urlPatterns={"/servlet/InitParamAnno"}
350 ,
351         initParams = { @WebInitParam ( name="dirPath", value = "d:\\test") ,
352                         @WebInitParam ( name="userid", value = "system") })
353 public class InitParamAnnoServlet extends HttpServlet {
354     protected void doGet(HttpServletRequest request, HttpServletResponse response)
355         throws ServletException, IOException {
356         System.out.println(getInitParameter("dirPath"));
357         System.out.println(getInitParameter("userid"));
358     }
359 }
```

-결과

```
d:\test
system
```

365 12. Servlet Lifecycle

366 1)init() --> service() --> destroy()

368 2)init()

369 -처음 Servlet이 load되면, 초기화 작업수행, File을 열거나, DB를 열거나...

370 -일단 Memory에 load된 후에는(init() 되면), Memory에 계속 남아있으면서 Client의 요청때마다 다시 reload 되는 것이 아니라, 처음 load 단 한번만 수행됨.

372 3)service()

373 -Client의 요청때마다 수행됨

374 -Thread가 생성되서 병행적으로 수행됨

375 -get방식으로 요청이 오면 doGet() 호출

376 -post방식으로 요청이 들어오면 doPost() 호출

378 4)destroy()

379 -Memory에서 Unload될 때 한번만 호출

380 -명시적 Unload

381 a. Servlet Container(Servlet Engine, tomcat)이 Stop할 때

382 b. Tomcat Manager 창에서 해당 Context 를 Reload버튼을 누를 때

383 -암시적 Unload

384 a. Code 수정했을 때

387 13. CallBackServlet.java

```
389 public class CallBackServlet extends HttpServlet {
390     int initCallCount = 1;
391     int serviceCallCount = 1;
392     int destroyCallCount = 1;
393     public void init(ServletConfig sc){
394         System.out.println("init() 호출 - " + initCallCount++);
395     }
396     public void service(ServletRequest req, ServletResponse res){
397         System.out.println("service() 호출 - " + serviceCallCount++);
398     }
399     public void destroy(){
400         System.out.println("destory() 호출 - " + destroyCallCount++);
401     }
}
```

```
402     }
403
404     -destroy() 는 tomcat service 를 stop 해야만 처리됨
405
406
407 14. CallbackServlet1.java
408
409     public class CallbackServlet1 extends HttpServlet {
410         private int initCallCount = 1;
411         private int serviceCallCount = 1;
412         private int destroyCallCount = 1;
413         private String fileName;
414         private FileOutputStream file;
415
416         public void init() throws ServletException {
417             try{
418                 fileName = getInitParameter("file");
419                 file = new FileOutputStream(fileName);
420                 file.write(String.valueOf("\r\ninit() Called --> " + initCallCount++).getBytes());
421             }catch(IOException ex){
422                 System.out.println(ex);
423             }
424         }
425         public void service(ServletRequest req, ServletResponse res){
426             try{
427                 file.write(String.valueOf("\r\nservice() Called --> " +
428                     serviceCallCount++).getBytes());
429             }catch(IOException ex){
430                 System.out.println(ex);
431             }
432         }
433         public void destroy() {
434             try{
435                 file.write(String.valueOf("\r\ndestroy() Called --> " +
436                     destroyCallCount++).getBytes());
437                 file.close();
438             }catch(IOException ex){
439                 System.out.println(ex);
440             }
441         }
442     }
443
444     <servlet>
445         <servlet-name>CallBack</servlet-name>
446         <servlet-class>CallBackServlet1</servlet-class>
447         <init-param>
448             <param-name>file</param-name>
449             <param-value>C:\temp\test.txt</param-value>
450         </init-param>
451     </servlet>
452
453 15. CronServlet.java
454     public class CronServlet extends HttpServlet {
455         private long num;
456         private Timer timer;
457         private TimerTask task;
```



```
458     public void init() {
459         int delay = 0;
460         int period = 5000;
461         timer = new Timer();
462         task = new TimerTask() {
463             public void run() {
464                 num++;
465             }
466         };
467         timer.scheduleAtFixedRate(task, delay, period);
468     }
469
470     public void destroy() {
471         task.cancel();
472         timer.cancel();
473     }
474
475     public void doGet(HttpServletRequest req, HttpServletResponse res)
476         throws ServletException, IOException {
477         res.setContentType("text/html;charset=UTF-8");
478         PrintWriter out = res.getWriter();
479         req.setCharacterEncoding("UTF-8");
480
481         out.println("<html>");
482         out.println("<head><title>크론 테스트</title></head>");
483         out.println("<body><h3>크론 테스트</h3>");
484         out.println("5초 간격으로 숫자가 증가함.");
485         out.println("새로고침 버튼을 눌러주세요");
486         out.println("<p>숫자 = " + num);
487         out.println("</body></html>");
488     }
489 }
490
491
492 16. Annotation 을 이용한 서블릿의 선처리 및 후처리 작업
493 1) @PostConstruct을 이용한 선처리 작업
494 - 서블릿의 init() 메소드가 호출되기 전에 수행되는 선처리 작업
495 @PostConstruct
496 public void postConstruct(){ }
497
498 2) @PreDestroy를 이용한 후처리 작업
499 - destroy()호출된 후에 수행되는 후처리 작업
500 @PreDestroy
501 public void cleanup(){ }
502
503 3) PostPreServlet.java
504
505 @WebServlet("/PostPre")
506 public class PostPreServlet extends HttpServlet {
507
508     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
509         ServletException, IOException {
510         System.out.println("doGet");
511     }
512
513     @Override
514     public void destroy() {
515         System.out.println("destroy");
516     }
517 }
```

```
515     }
516
517     @Override
518     public void init() throws ServletException {
519         // TODO Auto-generated method stub
520         super.init();
521         System.out.println("init");
522     }
523
524     @PostConstruct
525     public void initMethod(){
526         System.out.println("initMethod");
527     }
528
529     @PreDestroy
530     public void clean(){
531         System.out.println("clean");
532     }
533
534 }
535
536
537 17. Http Header Test
538
539 <html><head><title>HTTP 헤더</title></head>
540 <body>
541 <h2>헤더 테스트</h2>
542 <a href=/DemoWeb/HttpHeader>테스트</a>
543 </body>
544 </html>
545
546 -HttpHeaderServlet.java
547
548 public class HttpHeaderServlet extends HttpServlet {
549     public void service(HttpServletRequest req, HttpServletResponse res)
550         throws IOException, ServletException {
551
552         res.setContentType("text/html;charset=UTF-8");
553         PrintWriter out = res.getWriter();
554         req.setCharacterEncoding("UTF-8");
555
556         out.println("<html><head><title>헤더 정보</title></head>");
557         out.println("<body> <center><h2>헤더 정보</h2>");
558         out.println("</center><hr><ul>");
559
560         Enumeration e = req.getHeaderNames();
561         while(e.hasMoreElements()) {
562             String name = (String)e.nextElement();
563             String value = req.getHeader(name);
564             out.print("<li>");
565             out.print(name);
566             out.print(" : ");
567             out.print(value);
568         }
569         out.println("</ul></body></html>");
570     }
571 }
572
```

```
573 -ClientPullServlet.java
574     public class ClientPullServlet extends HttpServlet {
575         int count = 0;
576         protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
577             response.setContentType("text/html;charset=UTF-8");
578             PrintWriter out = response.getWriter();
579             response.setHeader("Refresh", "3");
580             out.println("<html>");
581             out.println("<head><title>Client Pull</title></head>");
582             out.println("<body>");
583             out.println("<h2>Client Pull 테스트</h2>");
584             out.println(++count);
585             out.println("</body>");
586             out.println("</html>");
587         }
588     }
589
590 -InfoServlet.java
591     public class InfoServlet extends HttpServlet {
592         protected void doGet(HttpServletRequest req, HttpServletResponse res) throws
            ServletException, IOException {
593             res.setContentType("text/html;charset=UTF-8");
594             PrintWriter out = res.getWriter();
595             out.println("<html>");
596             out.println("<head><title>Request 정보 출력</title></head>");
597             out.println("<body>");
598             out.println("<h1>요청정보출력</h1>");
599             out.println("<pre>");
600             out.println("Request method : " + req.getMethod());
601             out.println("Request URI : " + req.getRequestURI());
602             out.println("Request Protocol : " + req.getProtocol());
603             out.println("Servlet path : " + req.getServletPath());
604             out.println("Query String : " + req.getQueryString());
605             out.println("Server name : " + req.getServerName());
606             out.println("Server port : " + req.getServerPort());
607             out.println("Remote address : " + req.getRemoteAddr());
608             out.println("*****Request Header *****");
609             java.util.Enumeration enums = req.getHeaderNames();
610             while(enums.hasMoreElements()){
611                 String str = (String)enums.nextElement();
612                 out.println(str + " : " + req.getHeader(str));
613             }
614             out.println("</pre>");
615             out.println("</body></html>");
616         }
617     }
618
619
620 18. Refresh
621     1) res.setHeader("Refresh", "3;url=list");
622     2) out.println("<meta http-equiv='Refresh' content='3; url=list'>");
```