

2001-09 : WWW, HTTP, CGI,

2001-10 : (Java Servlet™)

2001-11 : JSP (JavaServer Pages™)

2001-12 : EJB (Enterprise JavaBeans™)

孔子曰 '知之者 不如好之者, 好之者 不如樂之者'

가 ,

가 ? 가 ? 가 ?

from Yongwoo ' s Park

HTTP

CGI

가

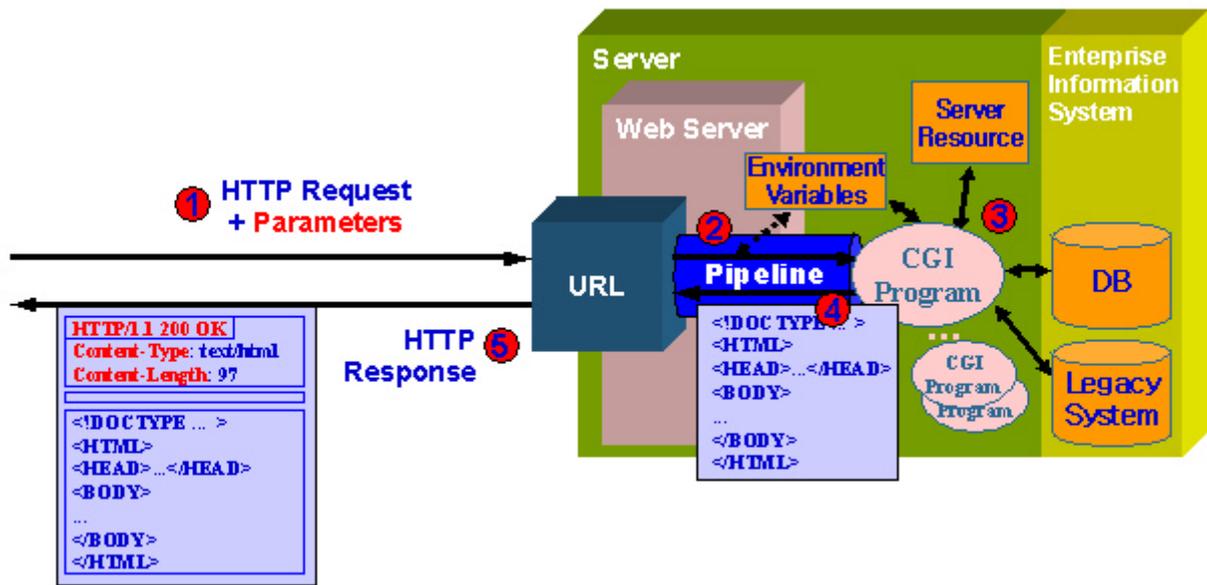
- 1. CGI VS. 4
 - 1.1 CGI 4
 - 1.2 , 5
 - 1.3 CGI vs. 7
- 2. 8
- 3. 三人三色: SERVLET, GENERICSERVLET, HTTPSERVLET.....11
- 4. 가 . HTTP 13
- 5. HTTP -.- ' 15
- 6. 가? () -.- ' 17
- 7. (TIP & TECH)..... 20
 - 7.1 가 FORWARD..... 20
 - 7.2 INCLUDE..... 21
 - 7.3 21
 - 7.4 22
 - 7.5 23
- 8. 24
- 9. 25

1. CGI vs.

1.1 CGI

CGI 가

1 HTTP HTTP CGI



1. HTTP CGI

HTTP (GET POST)

HTTP (

, HTML)

CGI CGI CGI CGI

CGI

(Pipeline)1

1 (A, B) , A B , A B 가

CGI 가 , 가 , 가 가 가 가

CGI MIME (text/html, HTML)

CGI , HTTP HTTP (Entity-Body)

CGI , CGI (Common Gateway Interface) HTTP

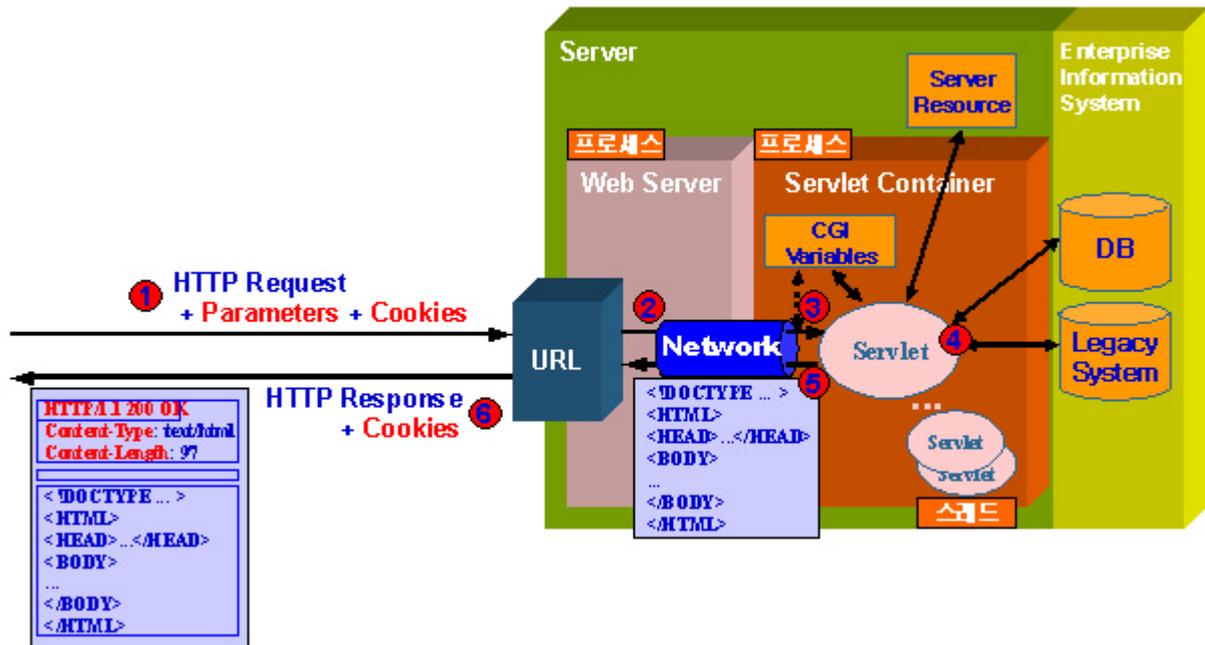
1.2

CGI 가 가 가

HTTP

CGI

가



2. HTTP

2

HTTP

,

“HTTP

CGI

”

가

가

가

1

,

,

가

3

가

Servlet

가

, JDBC

, HTTP

HTTP (GET, POST, ...)

HTTP 가 가

service() 가 init() 가 service() 가

HTTP

(Scalability) 가

1.3 CGI vs.

HTTP WWW CGI 가 가

URL, HTTP 가 URL, CGI

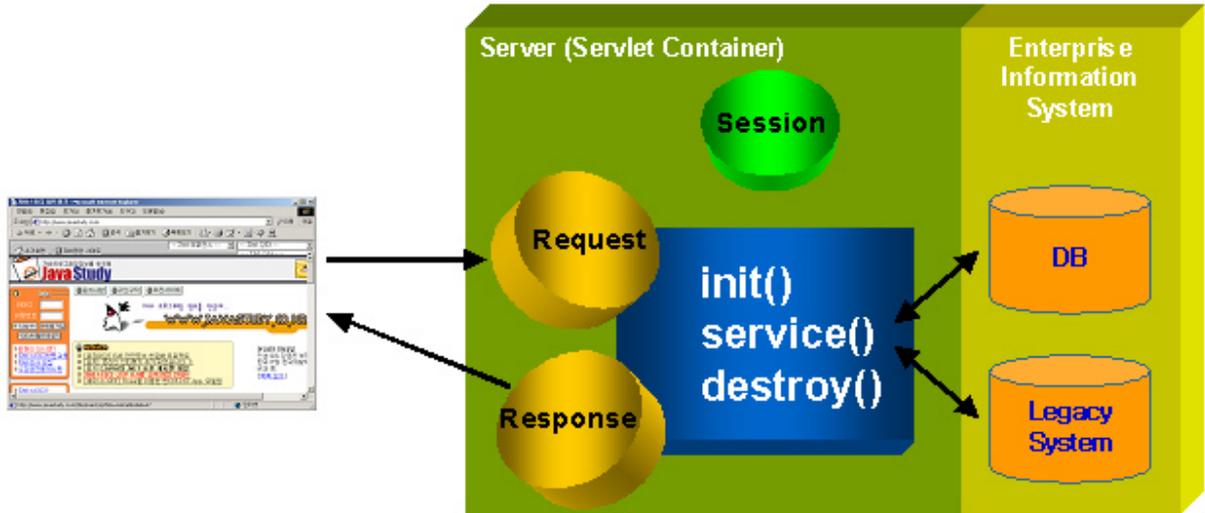
CGI (Response) 가

CGI

가 ,
.
CGI
.
()
가 가 ,
가 가
가 ,
가 ,
가

2.

2~3 API , “ 가가
!!” .
3
JSP
3
?



3.

3

가

()

(life-cycle)

public void init():

init()

init()

public void service(...):

init()

service(...)

ServletRequest

ServletResponse

service()

ServletRequest

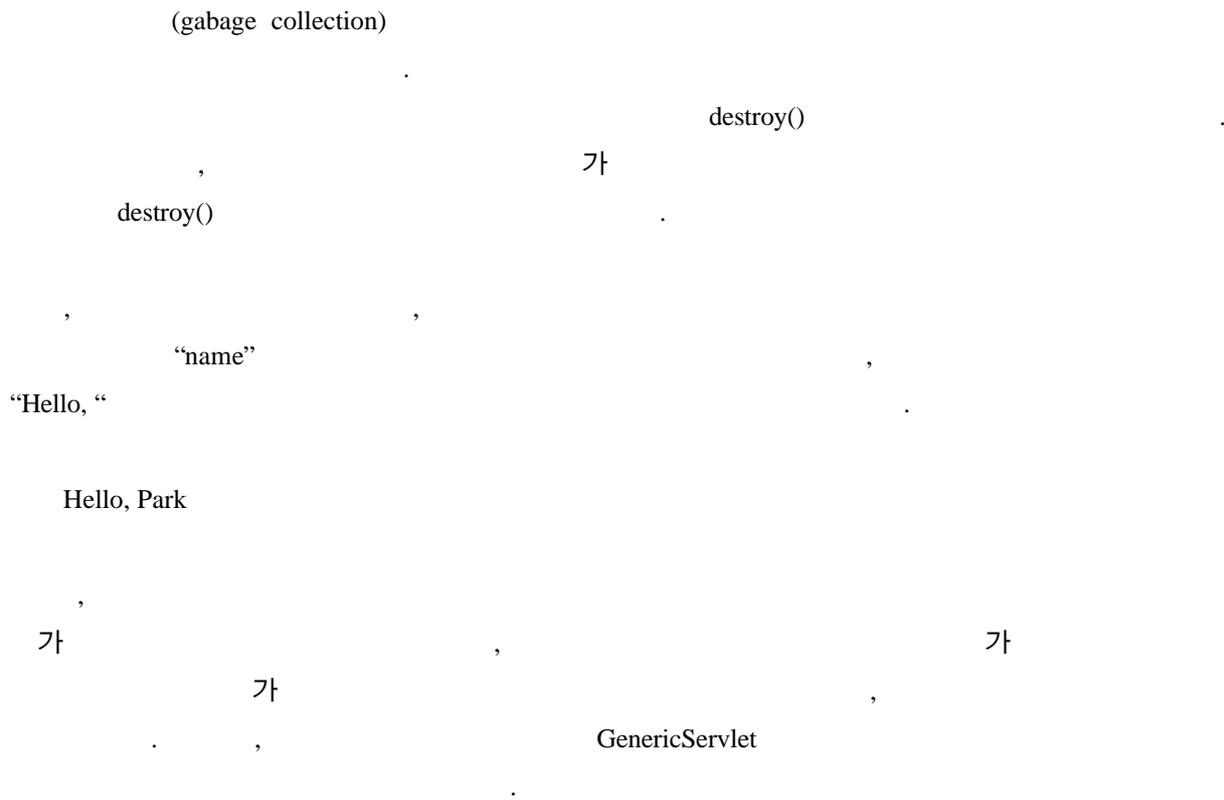
ServletResponse

PrintWriter

public void destroy():

가

가



```
import java.io.*;
import javax.servlet.*;

public class HelloServlet extends GenericServlet {
    public void init() {
        System.out.println("Created..."); //
    }

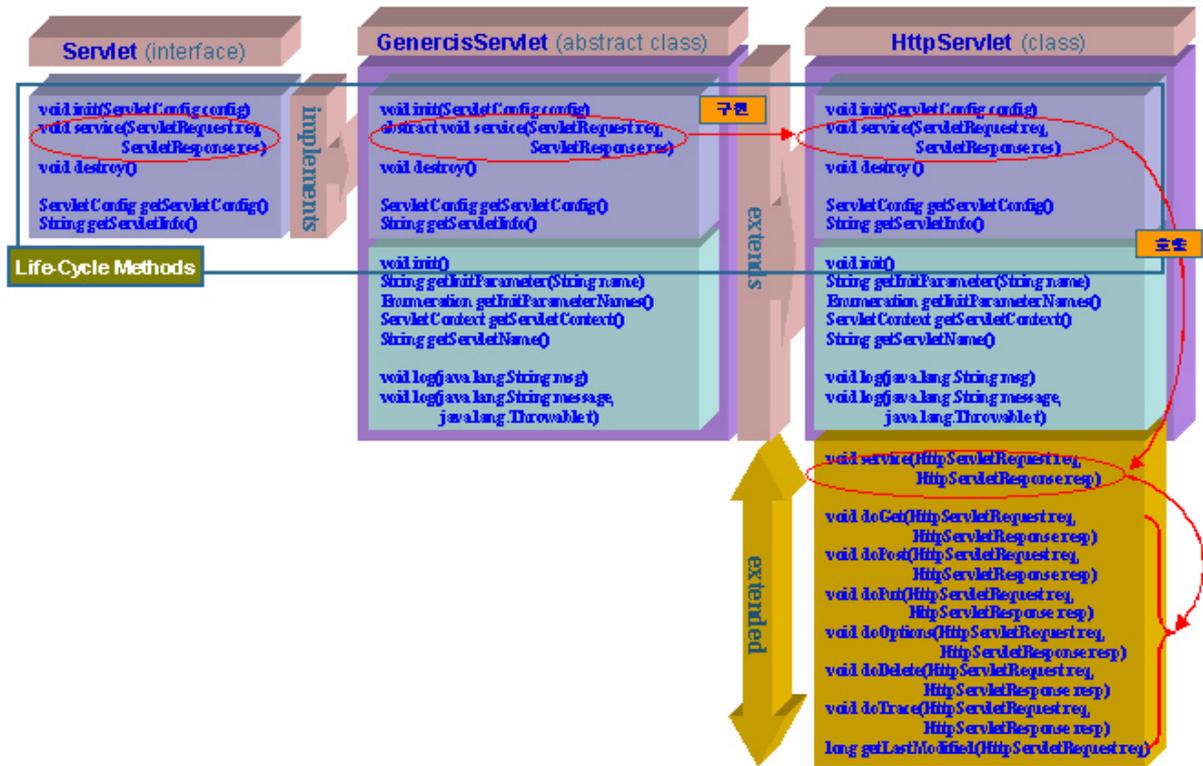
    public void destroy() {
        System.out.println("Destroyed..."); // 가
    }

    public void service(ServletRequest request,
        ServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html; charset=euc-kr");
        PrintWriter out = response.getWriter();

        String name = request.getParameter("name");

        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello, World!</title>");
        out.println("</head>");
        out.println("<body bgcolor=white>");
        out.println("<h1>Hello, " + name + "!</h1>");
        out.println("</body>");
        out.println("</html>");

        out.close();
    }
}
```

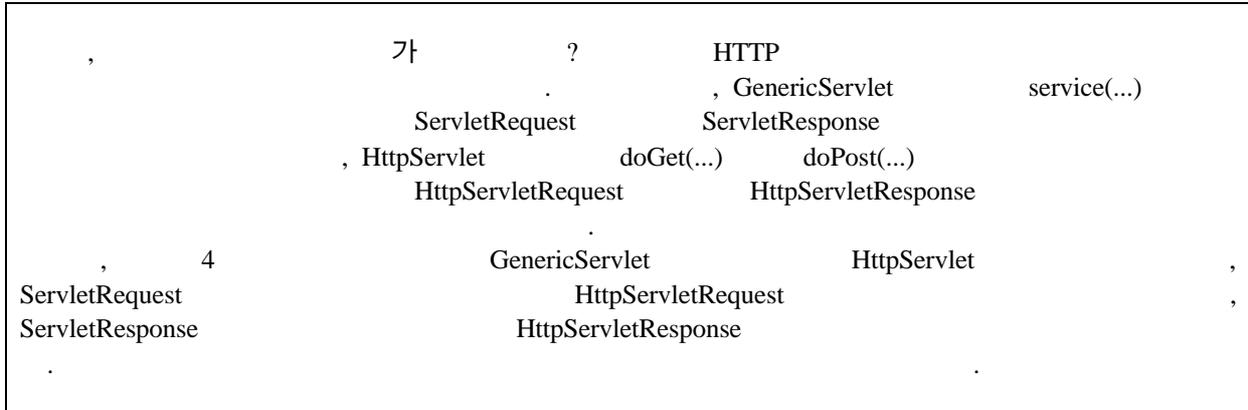



4. Servlet, GenericServlet, HttpServlet /

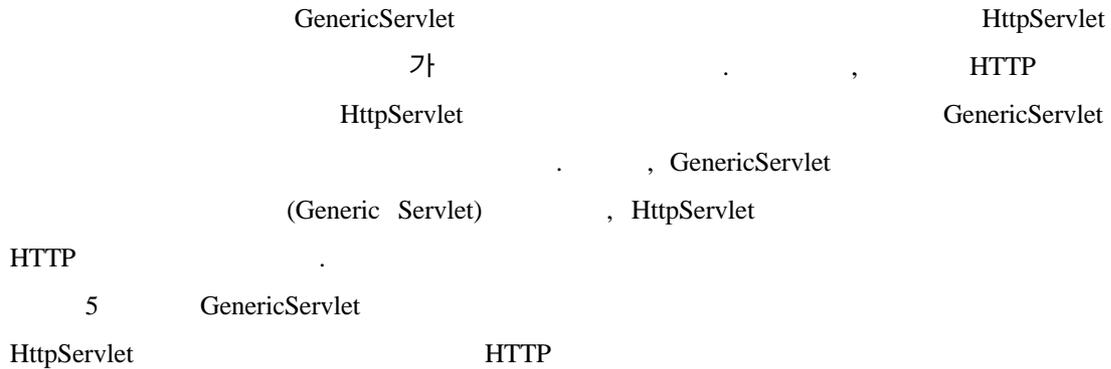
Servlet
 Servlet
 Servlet
 Servlet
 GenericServlet
 Servlet
 HTTP
 GenericServlet
 4
 GenericServlet
 HttpServlet
 Servlet
 GenericServlet
 service(...)
 service(...)
 GenericServlet
 가
 HttpServlet
 가
 , HttpServlet
 GenericServlet
 service(...)
 가

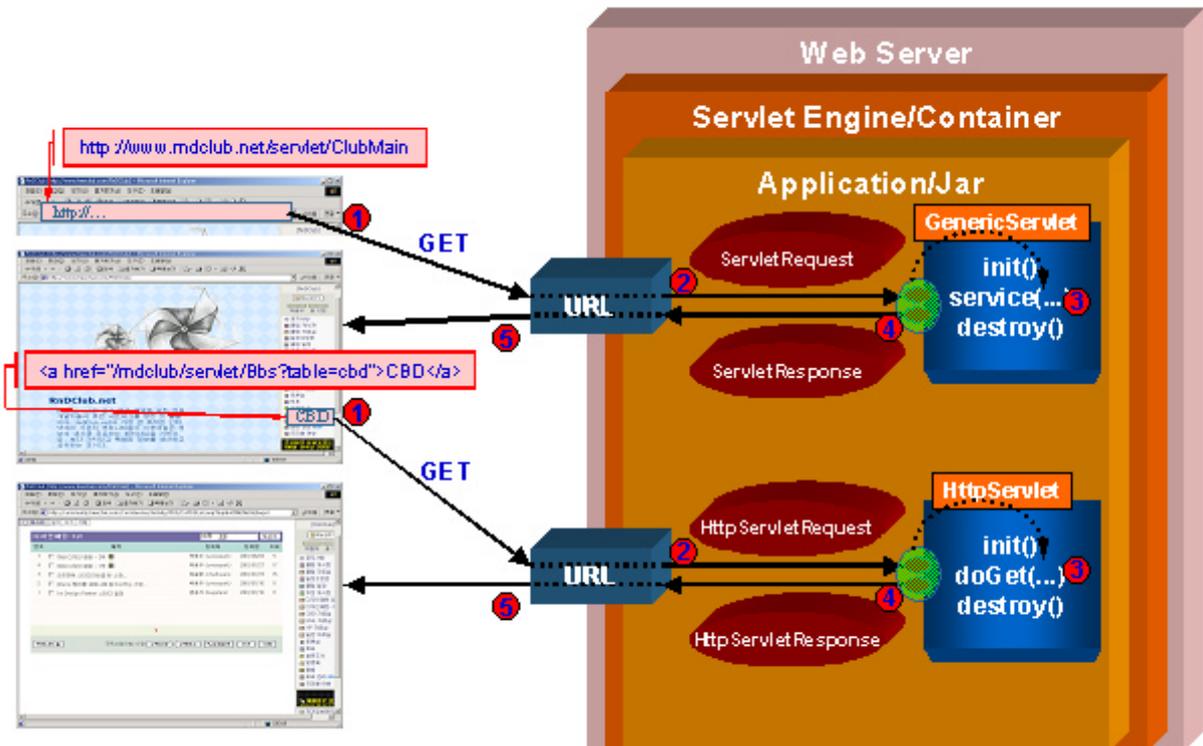
2 ServletRequest vs. HttpServletRequest

ServletResponse vs. HttpServletResponse



4. 가 . HTTP



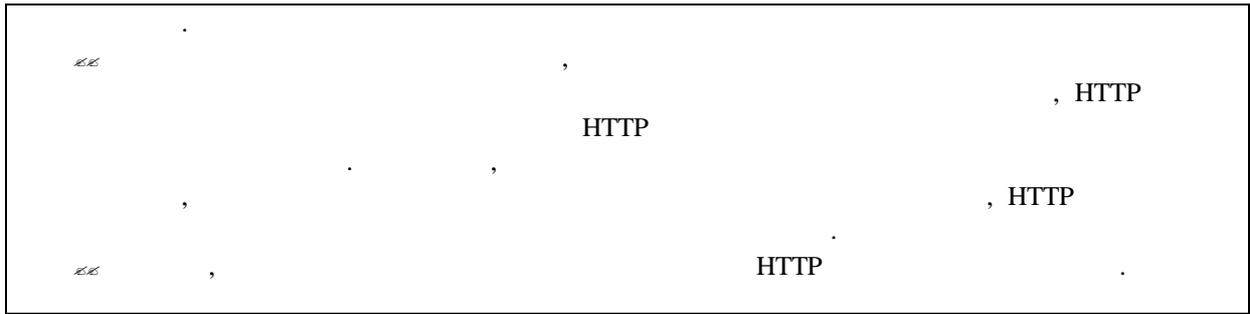


5. HTTP GET

5
 ,
 , ServletRequest ServletResponse
 service(...)
 ,
 HTTP ,
 HttpServletRequest HttpServletResponse HTTP doGet(...)
 doPost(...), HTTP
 , HTTP (Cookie) (Session)
 HTTP

3 vs. HTTP

HTTP
 ,
 // service(...) , HTTP doGet(...)
 doPost(...) doXxx(...) HTTP
 , HTTP
 service(...)
 HTTP
 // HTTP
 , 가



5. HTTP - . - ' .

, HTTP GET POST
 service(...) 가 .

```
public void service(ServletRequest request, ServletResponse response)
```

```
, HTTP  

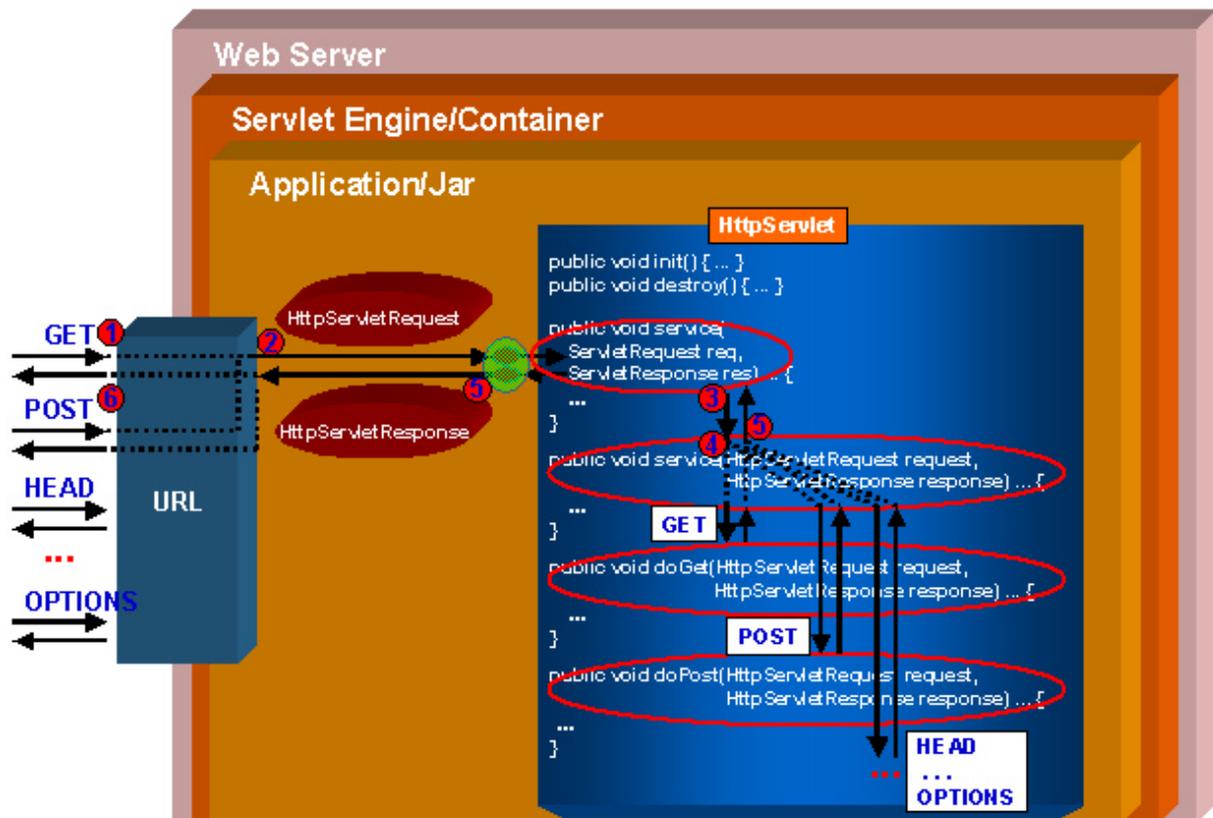
    . , HTTP HTTP  

    , HTTP GET  

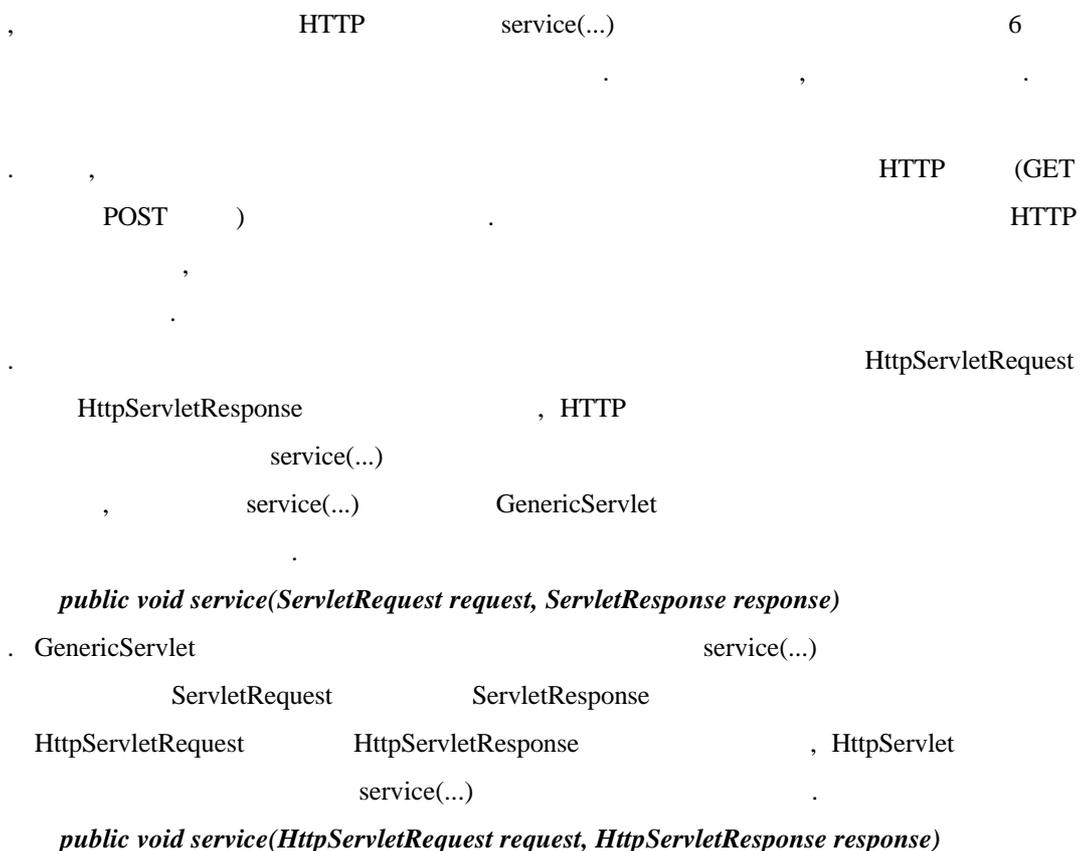
    doGet(...) , HTTP POST  

    doPost(...) . 6 HTTP  

    HTTP
```

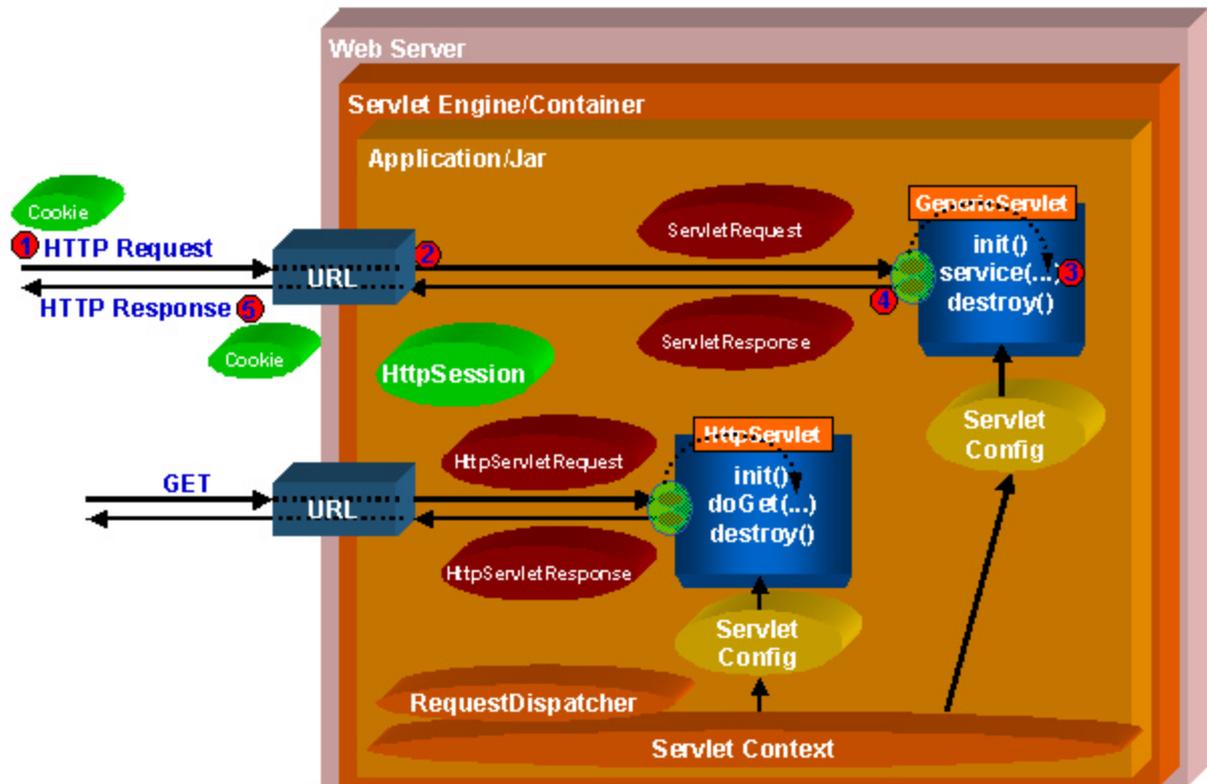


6. HTTP



Java Servlet Architecture

GenericServlet
 javax.servlet.http : HTTP
 , HttpServletRequest
 , HttpServletResponse , HttpSession , HttpServlet
 Cookie , HttpSessionEvent , HttpUtils



7.

GenericServlet : GenericServlet
 service(...)
 ServletRequest : , CGI
 ServletResponse : , ServletResponse
 PrintWriter

```

, HTTP

// HttpServlet : HTTP
// HTTP doXxx() HttpServlet
// HttpServletRequest : HTTP
// , CGI
// , HTTP

// HttpServletResponse :
// , ServletResponse
// PrintWriter ,
// HTTP 가

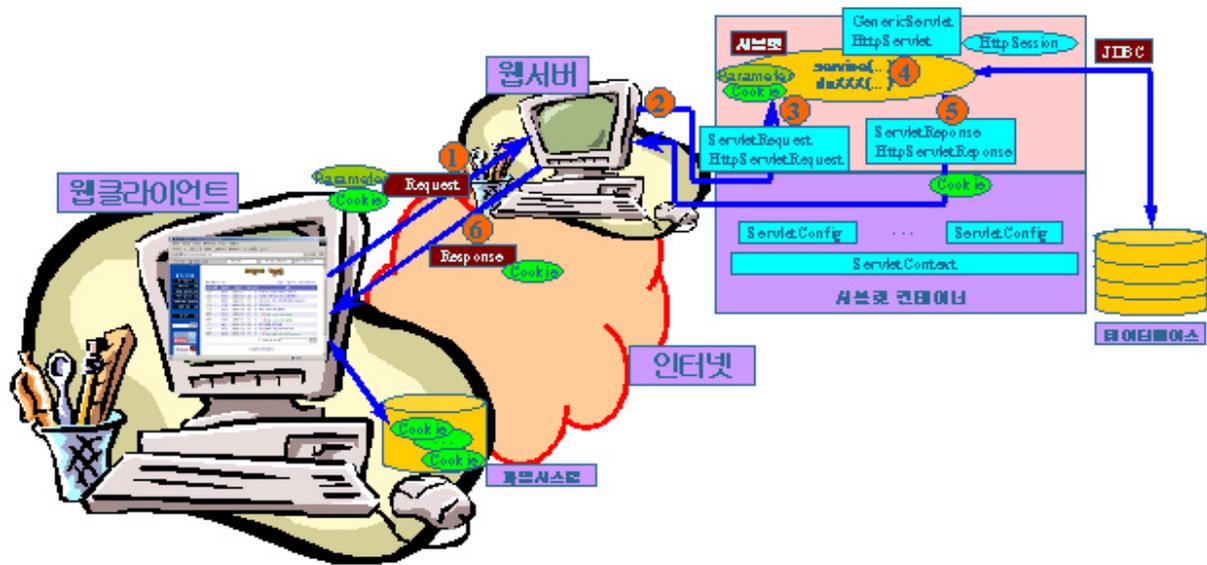
, HTTP

// javax.servlet.ServletConfig : HTTP
// javax.servlet.ServletContext : HTTP
// , conf
// , include forward
// RequestDispatcher
// javax.servlet.RequestDispatcher :
// include , forward
// RequestDispatcher
// javax.servlet.http.HttpSession : 가

가

// javax.servlet.http.Cookie :
,
    
```

HTTP (Cookie) (Session) HTTP HTTP



8.

7. (Tip & Tech)

&

7.1 가 forward

가

forward

```
try {
    ...
} catch(Exception e) {
    ServletContext context = getServletContext();
```

```
request.setAttribute("exception", e);
RequestDispatcher disp = context.getRequestDispatcher("/servlet/ErrorPageServlet");
disp.forward(request, response);
}
```

7.2

include

가 , . 가
가 ,
.
include , .

```
RequestDispatcher disp=null;
ServletContext context = getServletContext();
...
disp = context.getRequestDispatcher("/HeaderPage.html");
disp.include(request, response);
...
disp = request.getRequestDispatcher("/FooterPage.html");
disp.include(request, response);
```

7.3

가 ,
가 , 가 .
가?
가 가 , 가
가 가 ,
.

```
String author = request.getParameter("Author");
String email = request.getParameter("EMail");
```

```

Cookie cookie=null;
cookie = new Cookie("Author", author);
cookie.setMaxAge(365*24*3600); // 365 * 24 * 3600
response.addCookie(cookie);
...
cookie = new Cookie("EMail", email);
cookie.setMaxAge(365*24*3600); // 365 * 24 * 3600
response.addCookie(cookie);

```

Cookie

setMaxAge() -1

가

0

가

, 365

7.4

가

가

가

for

가

가

가

```

String login=null;
Cookie[] cookies = request.getCookies();
for(int i=0;i<cookies.length;i++) {
    if("login".equals(cookies[i].getName())) {
        login = cookies[i].getValue();
        break;
    }
}

```

가 Hashtable

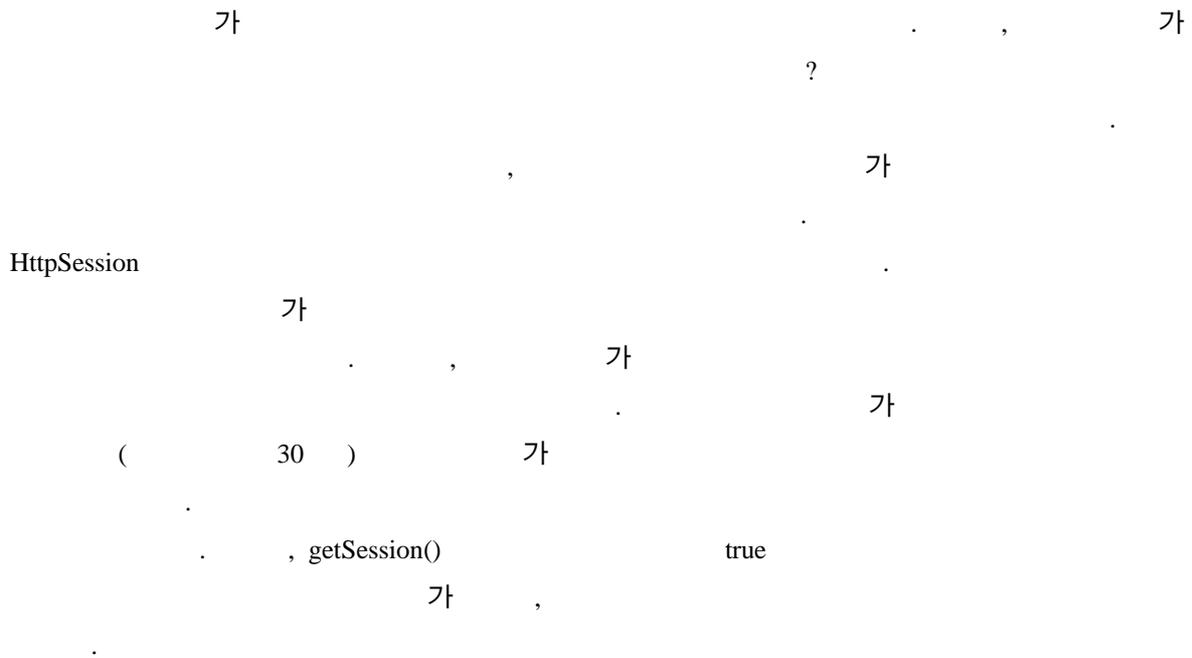
```

Hashtable cookieHash = new Hashtable();
Cookie[] cookies = request.getCookies();

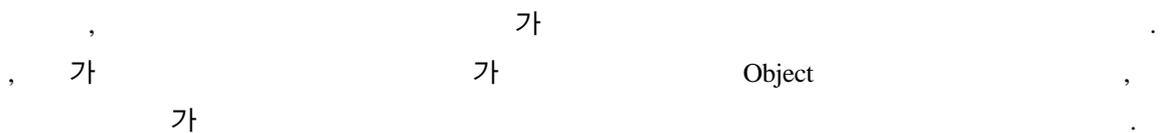
```

```
for(int i=0;i<cookies.length;i++) {  
    Cookie cookie = cookies[i];  
    String name = cookie.getName();  
    String value = cookie.getValue();  
    cookieHash.put(name, value);  
}  
String author = (String)cookieHash.get("author");  
String phone = (String)cookieHash.get("phone");  
...
```

7.5



```
String id = request.getParameter("id");  
String password = request.getParameter("password");  
...  
HttpSession session = request.getSession(true);  
session.setAttribute("login.id", id);  
session.setAttribute("login.password", password);
```



```
HttpSession session = request.getSession(true);  
...  
String id = (String)session.getAttribute("login.id");  
String password = (String)session.getAttribute("login.password");
```

```
    ,      가  
        가      .      isNew()
```

```
SessionBindingData data = null;  
HttpSession session = request.getSession(true);  
if(session.isNew()) {  
    data = new SessionBindingData();  
} else {  
    data = (SessionBindingData)session.getAttribute("mySessionData");  
    data.myCount++;  
}  
session.setAttribute("mySessionData", data);
```

8.

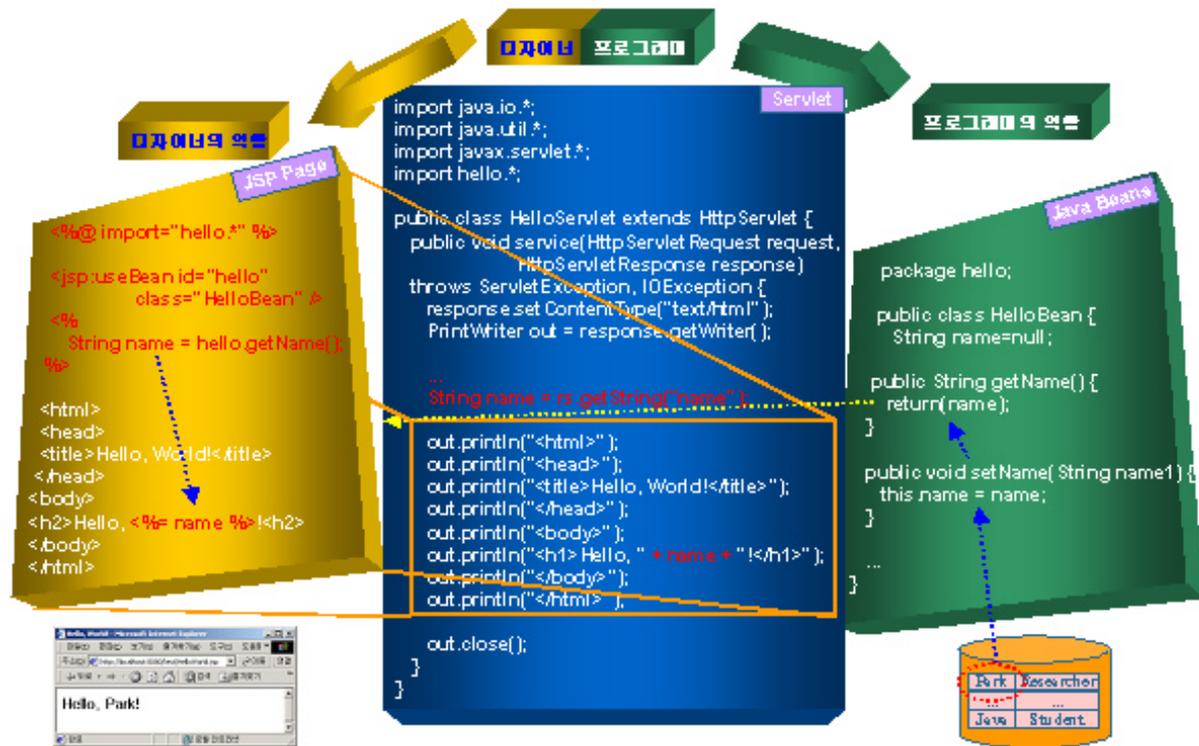
CGI

```
    ,      ,  
    가      가      ,
```

```
    ,      ,
```

```
    ,      9      HTML
```

```
    ,      , HTML
```



9.

9

HTML 가 JSP

가

?

9.

- [1] Java Web Application Programming Bible, ,
- [2] Java™ Servlet Specification v2.3
- [3] Java Servlet Programming, Jason Hunter with William Crawford, O'REILLY
- [4] Core Servlets and JavaServer Pages™, Marty Hall, Sun Microsystems
- [5] JavaServer Pages™ Software, Servlets, and Active Server Pages: Alternatives to CGI, James W. Cooper, JavaOne 2000
- [6] Using J2EE™ Technology: A Fast Introduction to Programming With the Java™ Servlet API, Marty Hall, JavaOne 2000
- [7] JavaServer Pages™ Technology and Java™ Servlet Components Overview, Danny Coward, Eduardo Pelegri-Llopert, JavaOne 2000
- [8] JavaServer Pages™ (JSP™) Technology and Servlets-Architecting an Enterprise Solution, Eduardo Pelegri-Llopert, Danny Coward, JavaOne 2000
- [9] What's New in the Java™ Servlet 2.2 API and the JavaServer Pages™ 1.1 API (JSP™), Arny Epstein, JavaOne 2000