Java Web Programming 입문

(Servlet/JSP)

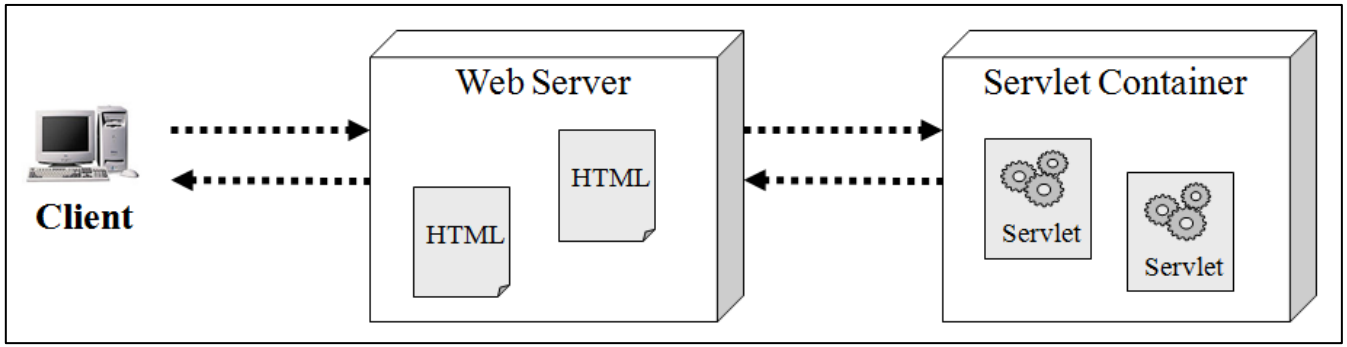
21일차

오늘의 키워드

- Servlet
  - Web Server
  - Servlet Container
  - Servlet 작성
  - Servlet 족보 (Servlet 상속관계)
  - Get
  - Post
  - HttpServletRequest
  - HttpServletResponse
  - Servlet의 한글 Encoding
  - HTTP Request Message Header
  - HTTP Response Message Header

Servlet

- Web Server
  - Servlet가 동작하기 위해서는
    - Web Server
      - HTTP 요청(Rquest)을 받고 응답(Response)
    - Servlet Container
      - Servlet 보관, 관리, 실행
  - Client의 Request를 처리 후 결과를 돌려줌
    - HTML 문서를 찾아 Client에게 전달해주는 일만 처리
    - Servlet에 대한 Request가 올 경우
      - Servlet Request 정보를 묶어 Servlet Container에 전달
      - Servlet 실행 후, 결과를 Client에게 전달



Servlet

- Servlet Container
  - Web Server에게서 Servlet 요청(Request)을 받으면
    - Request 정보를 이용하여
      - 요청(Request) 객체(Object) 생성
        - HttpServletRequest Object
      - 응답(Response) 객체(Objet) 생성
        - HttpServletResponse Object

```
HttpServletRequest  req = new HttpServletRequest();
HttpServletResponse res = new HttpServletRespone();
```

    - Servlet Object 생성
      - 요청된 Servlet이 Servlet Container에 잇는지 검사
        - Servlet Container에 Loading 되어 있다면
          - 해당 Servlet 객체 생성
        - Servlet Container에 Loading 되어있지 않다면
          - 해당 Servlet의 Class 파일을 Loading 후 Object 생성

```
MyServlet m = new MyServlet();
```

    - Servlet 실행
      - Servlet 작성시 HttpServlet Class를 상속
      - doGet(), doPost() Method는 재정의하여 Servlet 구현
      - 위 Method 들이 Servlet Object가 생성 후 Servlet Container에서 호출, 실행

```
m.doGet(req, res);
```

Servlet

- Servlet 작성
  - HttpSevlet Class 상속
  - Servlet Class 는 public 으로 선언

```
public class MyServlet
     extends HttpServlet {
          ...
}
```

  - doGet() or doPost() Method 재정의

```
public void doGet( HttpServletRequest req,
                   HttpServletResponse res) {
     ...
}
```


GET 방식 = 주소창을 이용해서 정보를 전달합니다


POST 방식 = HTTP 헤더 안에 넣어 정보를 전달합니다

Servlet

- Servlet 작성
  - doGet() Method 작성
    - setContentType() Method
      - HttpServletResponse (Response Object) 의 Method
      - Response Page의 MIME Type, Encoding Style 지정

```
public void setContentType(java.lang.String type)
```

```
res.setContentType("text/html;charset=euc-kr");
```

    - getWriter() Method
      - Client로의 Stream 얻기

```
PrintWriter out = res.getWriter();
```

    - HttpServletRequest (Request Object)의 정보 출력

```
out.println(req.getProtocol()    + "<br>");
out.println(req.getRemoteAddr() + "<br>");
out.println(req.getRemoteHost() + "<br>");
out.println(req.getScheme()      + "<br>");
out.println(req.getServerName() + "<br>");
out.println(req.getServerPort() + "<br>");
```

Servlet

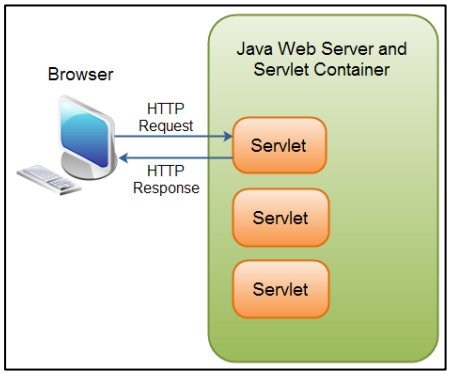
- Servlet 작성 (MyServlet.java)

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {
    public void doGet( HttpServletRequest req, HttpServletResponse res)
                        throws ServletException, IOException {
        PrintWriter out = res.getWriter();
        res.setContentType("text/html;charset=euc-kr");
        out.println("<html>                                    ");
        out.println("    <body>                               ");
        out.println(         req.getProtocol()    + "<br> ");
        out.println(         req.getRemoteAddr()  + "<br> ");
        out.println(         req.getRemoteHost()  + "<br> ");
        out.println(         req.getScheme()      + "<br> ");
        out.println(         req.getServerName()  + "<br> ");
        out.println(         req.getServerPort()  + "<br> ");
        out.println("    </body>                              ");
        out.println("</html>                                  ");
    }
}
```

Servlet

- Servlet 적용
  - MyServlet.java
  - …/webapps/MySample/WEB-INF/classes
  - Javac MyServlet.java
  - http://localhost:8080/MySample/servlet/MyServlet
  - http://127.0.0.1:8080/MySample/servlet/MyServlet

- Servlet의 동작
  - Servlet Container에서 Request 정보 전달
  - Request 를 처리할 Servlet 검색
  - Servlet이 없을경우 요청된 Servlet Object 생성
  - Request, Response Object 생성
  - Servlet 호출
  - Servlet 실행
  - Response Object를 통행 Client에게 결과 전송



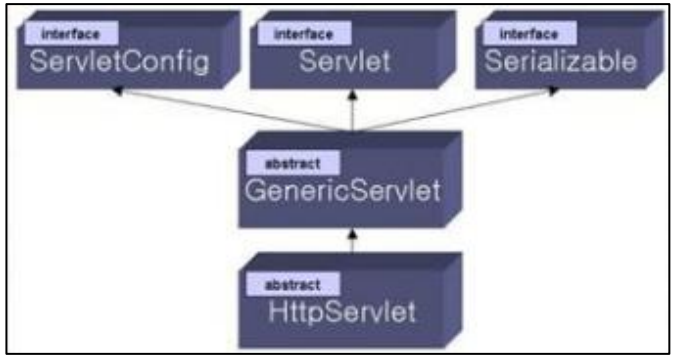
Servlet

- Servlet 족보
  - Servlet 구현 시 HttpServlet 상속
  - HttpServlet(Abstract Class) 은 GenericServlet 상속

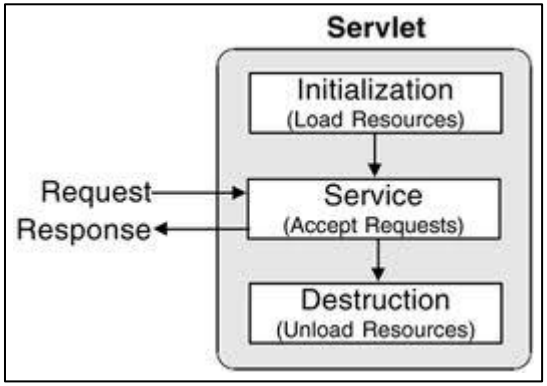
```
public abstract class HttpServlet
    extends      GenericServlet
    implements   java.io.Serializable {
    ...
}
```

  - GenericServlet(Abstract Class) 은
    - ServletConfig Interface 구현
    - Servlet Interface 구현
    - Serializable Interface 구현

```
public abstract class GenericServlet
    extends      java.lang.Object
    implements   Servlet,
                 ServletConfig,
                 java.io.Serializable {
    ...
}
```



Servlet



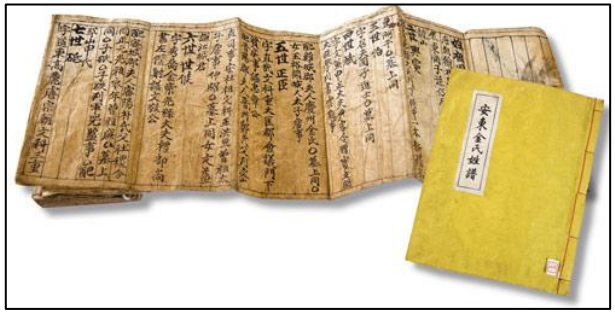
- Servlet 족보
  - ServletConfig Interface
    - Servlet과 관련된 초기화 정보 처리 시 필요한 Method 정의
  - Servlet Interface
    - Servlet 기능과 관련된 Method 정의

    - Servlet의 삶과 죽음
      - init()
        - Servlet이 처음 Loading 될 때 (Object가 생성될 때) 한번 호출되는 method
        - 처음 Servlet의 Request가 왔을때 Servlet Container가 Memory에 Loading
        - Servlet Object 생성과 동시에 init() Method 호출
        - 오류 시 UnavailableException or ServletException 발생
      - service()
        - Client로 부터 Servlet에 대한 Request가 있을 때 호출되는 method
        - Init() Method 호출 후, 수행되어 Client의 Request를 실행
        - Client의 Request가 있을 때마다 실행
        - Request 방식에 따라 doGet() or doPost() Method 호출

      - destroy()
        - Servlet이 Memory에서 제거될 때 호출되는 Method
        - Servlet Object가 더 이상 Service를 하지 않을 경우 사용
        - Memory에서 제거 될때 호출하는 method
        - 호출 후, Garbage Collector가 Servlet Object를 Memory에서 제거

Servlet

- Servlet 족보
  - Serializable Interface
    - Servlet Object의 직렬화를 위한 Interface
  - GenericServlet Class
    - ServletConfig, Servlet, Serializable Interface를 구형
    - Abstract Class
    - 일반적인 Network Protocol에서의 Servlet 구현
    - 상속해서 사용 (Abstract Class)
    - Servlet, Servletconfig Inteface는 Servlet 통신에 필요한 기본적인 Metho들을 가지고 있음
      - GenericServlet에서 구현
    - HTTP Protocol 이외 Protocol에서 동작하는 Servlet
  - HttpServlet Class
    - GenericServlet Class를 상속
    - HTTP Protocol에 최적화된 Class
    - WWW(World Wide Web) – HTTP

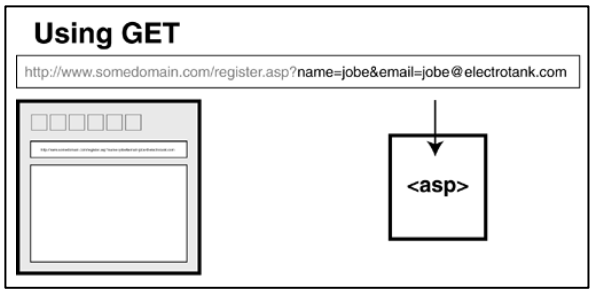


Servlet

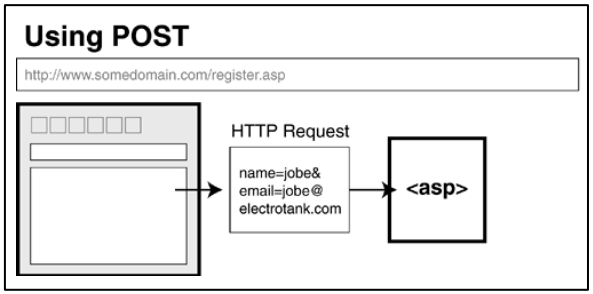
- HttpServlet Class
  - GenericServlet Class는 일반적인 Network Protocol용
  - HTTP Protocol에 맞게 구현된 Class
  - Web Client 요청 처리 후, 결과 return
  - doGet() or doPost() Method를 재정의해서 사용
  - Servlet의 service() Method 호출 후
    - Client의 요청이 GET / POST 인지에 따라
      - doGet(), doPost() Method가 호출
  - init(), service(), destroy()
    - HttpServlet Class는 service() Method 재정의하지 않음
    - service() Method는 자동으로 호출

Servlet

- GET VS POST
  - GET
    - 전달 정보를 URL에 포함하여 호출
    - 형식
      - Address?param1=var1&param2=var2…paramN=varN
      - http://localhost/MySample/servlet/HttpGetServlet?name=Love&age=30
      - URL에 정보가 노출 : 보안 취약
      - 보내는 정보 크기 1024byte로 제한



  - POST
    - 전달 정보를 첨부파일 형태로 포함하여 전달
    - URL 로는 정보 열람이 불가
      - URL에 정보가 비노출 : 보안 좋음
    - 보내는 정보의 크기에 제한 없음



Servlet

- doGet()
  - doGet) Method 재정의

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

```
public void doGet ( HttpServletRequest req,
                    HttpServletResponse res )
            throws ServletException, IOException {
    …
}
```

  - getParameter() Method
    - Servlet에 전달된 Parameter 정보
    - Servlet에 전달된 Parameter 이름을 인자로 가짐

```
public java.lang.String getParameter(java.lang.String name)
```

```
String helloName = req.getParameter("name");
```

Servlet

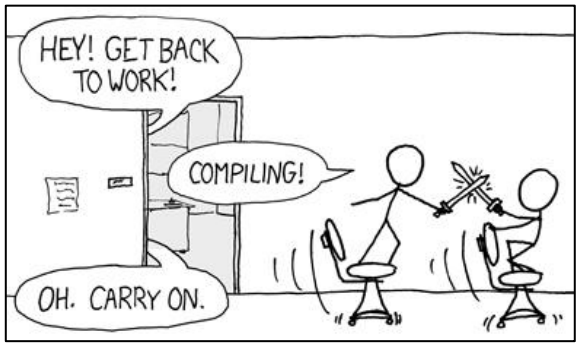
- HttpGetServlet.java

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class HttpGetServlet extends HttpServlet{

    public void doGet ( HttpServletRequest req,
                        HttpServletResponse res )
                throws ServletException, IOException{

        PrintWriter out = res.getWriter();
        String helloName = req.getParameter("name");

        out.println("<html>                                ");
        out.println("    <body>                            ");
        out.println("        <h1>doGet Method</h1>         ");

        if(!helloName.equals(""))
            out.println("    <h2>"+ helloName +"</h2>");

        out.println("    </body>                           ");
        out.println("</html>                               ");
        }
}
```

- .../webapps/MySample/WEB-INF/classes
- javac HttpGetServlet.java

Servlet

- Get 방식으로 Servlet을 호출하는 HTML 작성
  - <form> Tag
    - Client에게 정보를 받아 Server로 전달할 때 사용되는 Tag
    - 속성
      - action
        - Client 에게 받은 정보를 처리할 Servlet을 지정
      - method
        - Servlet에 정보를 전달할 방식을 지정
        - GET, POST 중 하나




```
<form action ="/MySample/servlet/HttpGetServlet" method="GET">
    ...
</form>
```

  - <input> Tag
    - Interactive Web Page 요소
      - Text Box, Button 등등을 만들기 위한 Tag
    - 속성
      - type
        - <input> Tag의 종류 지정
        - text (텍스트 상자), submit (전송 Button) 등등
      - name
        - <input> Tag의 이름, Parameter 이름으로 사용
      - value
        - <input> Tag로 Button을 만들때 Button에 표시될 Text 를 지정

```
<input type="text" name="name">
<input type="submit" value="Submit">
```

Servlet

- ExcuteGetServlet.html

```
<HTML>
    <HEAD>
        <TITLE>서블릿의 doGet</TITLE>
    </HEAD>
    <BODY>
        <h1>GET 방식전송</h1>
        <form action ="/MySample/servlet/HttpGetServlet" method="GET">
            <input type="text" name="name" >
            <input type="submit" value="submit">
        </form>
    </BODY>
</HTML>
```

- 적용 & 실행
  - …/webapps/MySample/
  - http://localhost:8080/MySample/ExcuteGetServlet.html
  - http://127.0.0.1:8080/MySample/ExcuteGetServlet.html

  - http://localhost:8080/MySample/servlet/HttpGetServlet?name=Hahaha
  - http://127.0.0.1:8080/MySample/servlet/HttpGetServlet?name=Hahaha

Servlet

- doPost()
  - doPost() Method 재정의

```
public void doPost ( HttpServletRequest request,
                     HttpServletResponse response )
            throws ServletException, IOException
```

```
public void doPost ( HttpServletRequest req,
                     HttpServletResponse res )
            throws ServletException, IOException {
    …
}
```

    - Client에서 전달된 HTTP 메시지가 POST 방식일때 사용

  - Parameter 값 추출

```
String helloName = req.getParameter("name");
```

Servlet

- HttpPostServlet.java

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class HttpPostServlet extends HttpServlet{
    public void doPost( HttpServletRequest req,
                        HttpServletResponse res )
                throws ServletException, IOException{

        PrintWriter out = res.getWriter();
        String helloName = req.getParameter("name");
        out.println("<h1>doPost Method</h1>");
        out.println("<h2>"+ helloName +"</h2>");
    }
}
```

  - …/webapps/MySample/WEB-INF/classes
  - Javac HttpPostServlet.java

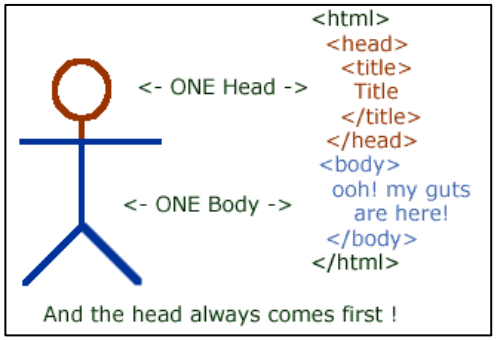
Servlet

- Post 방식으로 Servlet을 호출하는 HTML 작성
  - <form> Tag

```
<form action ="/MySample/servlet/HttpPostServlet" method="POST">
    …
</form>
```

  - <input> Tag

```
<input type="text" name="name" >
<input type="submit" value="Submit">
```



Servlet

- ExcutePostServlet.html

```
<HTML>
    <HEAD>
        <TITLE>서블릿의 doPost</TITLE>
    </HEAD>
    <BODY>
        <h1>POST 방식전송</h1>
        <form action ="/MySample/servlet/HttpPostServlet" method="POST">
            <input type="text" name="name" >
            <input type="submit" value="submit">
        </form>
    </BODY>
</HTML>
```

- 적용 & 실행
  - …/webapps/MySample/
  - http://localhost:8080/MySample/ExcutePostServlet.html
  - http://127.0.0.1:8080/MySample/ExcutePostServlet.html

Servlet

- Servlet 작업의 대부분은
  - HttpServletRequest Object
  - HttpServletResponse Object
- doGet(), doPost() Method의 매개변수
  - Client의 요청이 전달되면
    - Servlet Container 는 service() Method 호출
    - Client의 요청과 응답에 해당하는
      - HttpServletRequest, HttpServletResponse Object를 매개변수 형식으로 넘겨줌
- Servlet 내 핵심 Method

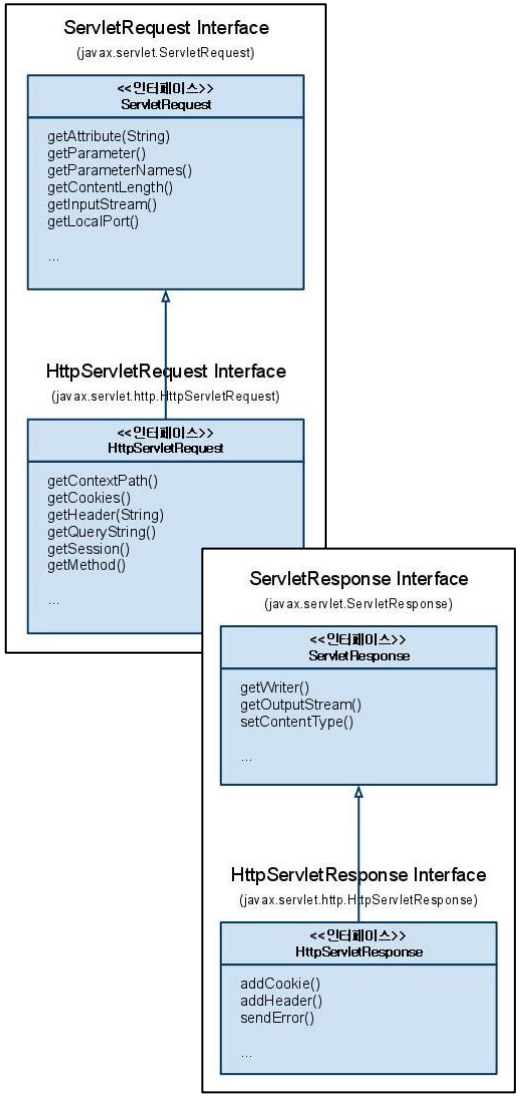
```
protected void service   (HttpServletRequest req, HttpServletResponse resp)

protected void doGet     (HttpServletRequest req, HttpServletResponse resp)

protected void doPost    (HttpServletRequest req, HttpServletResponse resp)
```

- 요청 응담 Object
  - Client의 요청 : HttpServletRequest Class
  - Client에 응답 : HttpServletResponse Class

Servlet

- HttpServletRequest Object
  - Client의 모든 요청 정보
  - 요청 HTTP Header 정보
  - Servlet으로 전달된 Parameter 정보
  - Client에서 전송된 InputStream 형태의 Data
  - Session, Cookie, etc…

- HttpServletResponse Object
  - Client에 전송할 응답 정보
  - 응답 HTTP Header 정보
  - Client로 전송될 OutputStream 형태의 Data
  - Session, Cookie, etc…



Servlet

- SimpleResponse.java

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class SimpleResponse extends HttpServlet{

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

        PrintWriter out = response.getWriter();
        String helloName = request.getParameter("name");
        out.println("<h2>Hello "+ helloName +"</h2>");
    }
}
```

  - .../webapps/MySample/WEB-INF/classes
  - javac SimpleResponse.java

Servlet

- SimpleRequest.html

```
<HTML>
    <HEAD>
        <TITLE>간단한 request와 response</TITLE>
    </HEAD>
    <BODY>
        <h2>GET 방식 요청</h2>
        <form action ="/MySample/servlet/SimpleResponse" method="GET">
            <input type="text" name="name" >
            <input type="submit" value="submit">
        </form>
    </BODY>
</HTML>
```

  - http://localhost:8080/MySample/SimpleRequest.html
  - http://127.0.0.1:8080/MySample/SimpleRequest.html

Servlet

- 한글이 깨져요! ㅠ_ㅠ
  - Servlet 에서의 Request, Response 메시지가 한글 처리가 가능하도록 Encoding 설정이 필요
  - Response Object의 한글 Encoding
    - setContentType() Method

```
public void setContentType(java.lang.String type)
```

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

Servlet

- ServletEncoding.java

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class ServletEncoding extends HttpServlet{

    public void doGet ( HttpServletRequest request,
                        HttpServletResponse response )
                    throws ServletException, IOException {
        process(request, response);
    }

    public void doPost ( HttpServletRequest request,
                         HttpServletResponse response )
                    throws ServletException, IOException {
        process(request, response);
    }

    public void process ( HttpServletRequest request,
                          HttpServletResponse response ) {
        request.setCharacterEncoding("euc-kr");
        response.setContentType("text/html;charset=euc-kr");
        PrintWriter out = response.getWriter();
        String helloName = request.getParameter("name");
        out.println("<html><body>");
        out.println("<h1>Servlet Encoding Result</h1>");
        out.println("<h2>"+ helloName +"</h2>");
        out.println("</body></html>");
    }
}
```

- …/webapps/MySample/WEB-INF/classes
- javac ServletEncoding.java

Servlet

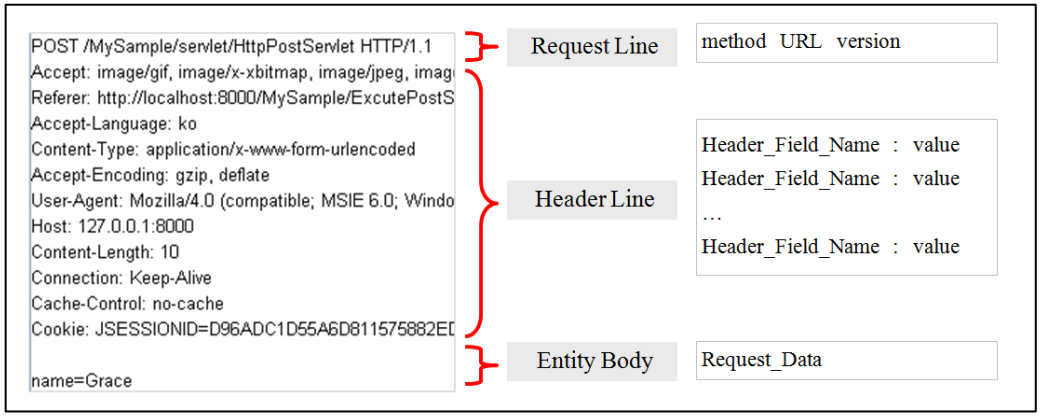
- ServletEncoding.html

```
<HTML>
    <HEAD>
        <TITLE>서블릿의 한글 인코딩</TITLE>
    </HEAD>
    <BODY>

        <h2>GET 방식전송</h2>
        <form action ="/MySample/servlet/ServletEncoding" method="GET">
            <input type="text" name="name" >
            <input type="submit" value="submit">
        </form>

        <h2>POST 방식전송</h2>
        <form action ="/MySample/servlet/ServletEncoding" method="POST">
            <input type="text" name="name" >
            <input type="submit" value="submit">
        </form>

    </BODY>
</HTML>
```

  - http://localhost:8080/MySample/ServletEncoding.html
  - http://127.0.0.1:8080/MySample/ServletEncoding.html

Servlet

- Http Request Message



  - Requetst Line
    - 요청 방식, URL
    - Method URL Version
      - POST, …MySample/servlet/HttpPostServlet, HTTP/1.1
  - HTTP Header Line
    - HTTP Protocol 이용 시 필요한 부가 정보 기술
    - Header_Field_Name : Value
      - HOST : 127.0.0.1:8080
  - Entity Body
    - Server로 전달하는 실제 Message 포함 부분

Refer	다른 페이지를 요청할 경우 이전 페이지의 주소
Accept-Language	클라이언트가 처리할 수 있는 언어
Content-Type	요청 문서의 미디어 타입
User-Agent	클라이언트의 요청 에이전트 종류
Host	서버의 기본 URL
Content-Length	전송되는 메시지의 길이

Servlet

- HTTP Request Message Header
  - HTTP Header 이름 집합
    - Enumeration Interface

```
Enumeration headerNames = request.getHeaderNames();
```

    - HTTP Header 이름 집합에서 HTTP Header 값 추출

```
while(headerNames.hasMoreElements()) {
    String name = (String)headerNames.nextElement();
    String value = request.getHeader(name);
}
```

    - HttpServletRequest Interface의 getHeader() Method

```
java.lang.String getHeader(java.lang.String name)
```

메소드 원형	메소드의 기능
String getContentType()	요청에 포함되어 있는 내용에 대한 MIME 타입을 반환, 모를 경우에는 null을 반환
int getContentLength()	요청에 포함되어 있는 데이터의 길이를 구하며, 만약 길이를 알 수 없는 경우에는 -1
String getMethod ()	HTTP의 요청 방식 출력
String getRequestURI()	요청 URL 중 프로토콜부터 쿼리 문자열까지의 부분을 반환
String getProtocol()	"HTTP/1.1" 과 같은 형식으로 프로토콜 및 major/minor 버전을 반환
String getAuthType()	서버에서 사용하는 인증 방법의 이름을 반환
String getRemoteUser()	사용자가 HTTP 인증을 통해 로그인 했을 경우, 사용자의 이름을 반환
String getQueryString()	요청 URL로부터 Query 문을 출력

Servlet

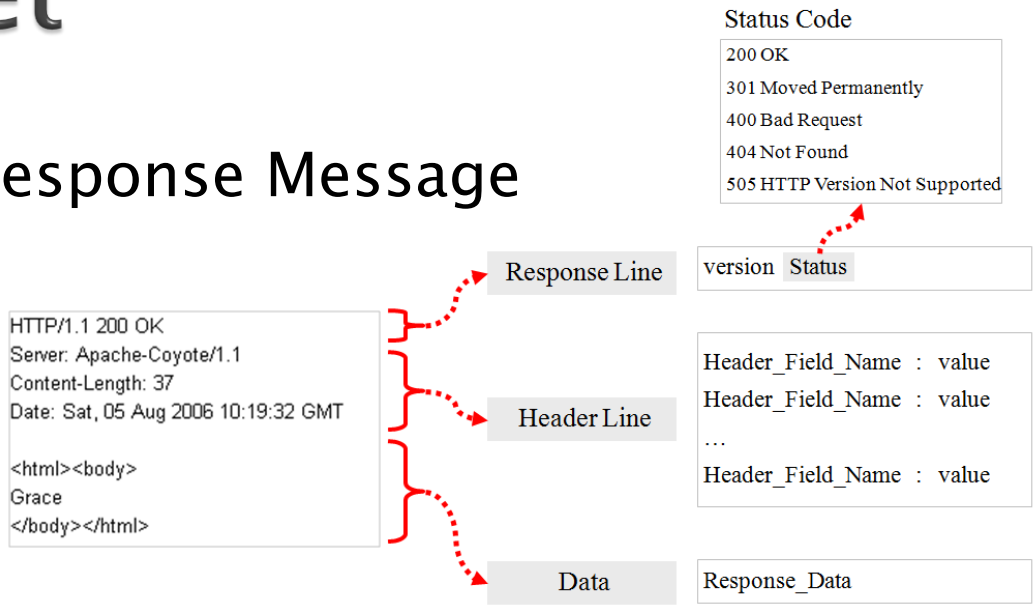
- PrintHeader.java

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

public class PrintHeader extends HttpServlet{

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

        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        Enumeration headerNames = request.getHeaderNames();
        while(headerNames.hasMoreElements()) {
            String name = (String)headerNames.nextElement();
            String value = request.getHeader(name);
            out.println(name + " : " + value + "<br>");
        }
        out.println("</body></html>");
    }
}
```

http://localhost:8080/MySample/servlet/PrintHeader
http://127.0.0.1:8080/MySample/servlet/PrintHeader

Servlet

- HTTP Response Message

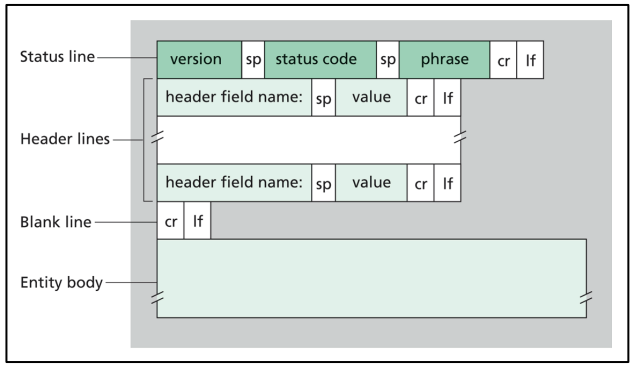


  - Response Line
    - Version, Status
      - HTTP/1.1 200 OK

상태 코드	설명
200 OK	요청이 성공적으로 처리되어 전달됨
301 Moved Permanently	URL이 옮겨짐
400 Bad Request	요청이 잘못됨
403 Forbidden	요청에 대한 수행이 거절됨
404 Not Found	요청한 파일이 존재하지 않음
500 Internal Server Error	예상하지 못한 서버처리 오류
503 Service Unavailable	서버부하로 요청에 대한 응답 불가

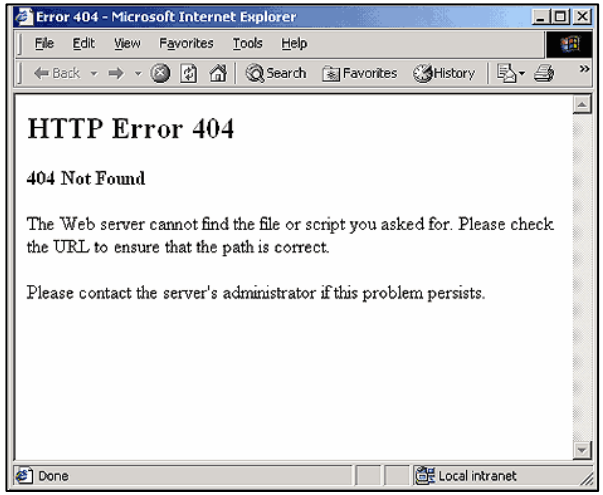
Servlet

- HTTP Response Message Header 항목
  - Cache-Control
    - Client가 받은 문서 Cache에 저장 여부 설정
  - Connection
    - Web Server 응답 완료 후 연결 유지 여부 설정
  - Expires
    - Cache에 문서 저장 시간 설정
  - Refresh
    - Web Browser 가 Refresh 항목에 지정된 Web Page 자동 연결 하도록 지정
  - Set-Cookie
    - Client에 일부 값을 저장후 사용하는 Cookie 설정



Servlet

- Response Status Code
  - Status Code
    - Client에게 전달된 요청 처리 결과 코드
    - http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
    - HttpServletResponse Interface에 멤버변수로 정의 (public static final)
  - Status Code Number
    - 200 ~
      - Client의 요청 수행 성공
    - 300 ~
      - File 이동 시 사용 Code
      - Location Header (File의 이동 위치)
    - 400 ~
      - Client의 요청 에러 (잘못된 요청)
    - 500~
      - Web Server에서 Client의 요청 처리중 Error 발생



Servlet

- Status Code Example
  - status parameter 값 저장

```
int status = Integer.parseInt(request.getParameter("status"));
```

  - Parameter 값에 따라 Error Page 전송
    - sendError() Method (HttpServletResponse)
      - status 값에 따라 상태코드 설정

```
void sendError(int sc)
```

```
switch(status) {
    case 400 :
        response.sendError(HttpServletResponse.SC_BAD_REQUEST);
        break;
    case 404 :
        response.sendError(HttpServletResponse.SC_NOT_FOUND);
        break;
    case 500 :
        response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
        break;
    default :
        response.sendError(HttpServletResponse.SC_OK);
}
```

Servlet

- StatusCode.java

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

public class StatusCode extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        int status = Integer.parseInt(request.getParameter("status"));
        switch (status) {
        case 400:
            response.sendError(HttpServletResponse.SC_BAD_REQUEST);
            break;
        case 404:
            response.sendError(HttpServletResponse.SC_NOT_FOUND);
            break;
        case 500:
            response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
            break;
        default:
            response.sendError(HttpServletResponse.SC_OK);
        }
    }
}
```

http://localhost:8080/MySample/servlet/StatusCode?status=에러코드
http://127.0.0.1:8080/MySample/servlet/StatusCode?status=에러코드

Servlet

- Refresh Header 설정
  - Response Message HTTP Header 항목 작성
    - HttpServletResponse Object
      - setHeader() Method 이용
        - Client로 전달될 응답 메시지 HTTP Header 설정

```
void setHeader(java.lang.String name, java.lang.String value)
```

```
response.setHeader("Refresh", "5;URL=http://www.naver.com/");
```

Servlet

- RefreshPage.java

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

public class RefreshPage extends HttpServlet{

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

        response.setContentType("text/html;charset=euc-kr");
        PrintWriter out = response.getWriter();
        response.setHeader("Refresh", "5;URL=http://www.naver.com/");
        out.println("<html><body>");
        out.println("<h2>5초 후 이동합니다.</h2>");
        out.println("</body></html>");
    }
}
```

http://localhost:8080/MySample/servlet/RefreshPage
http://127.0.0.1:8080/MySample/servlet/RefreshPage

오늘 숙제

- 집에서 해보자!