

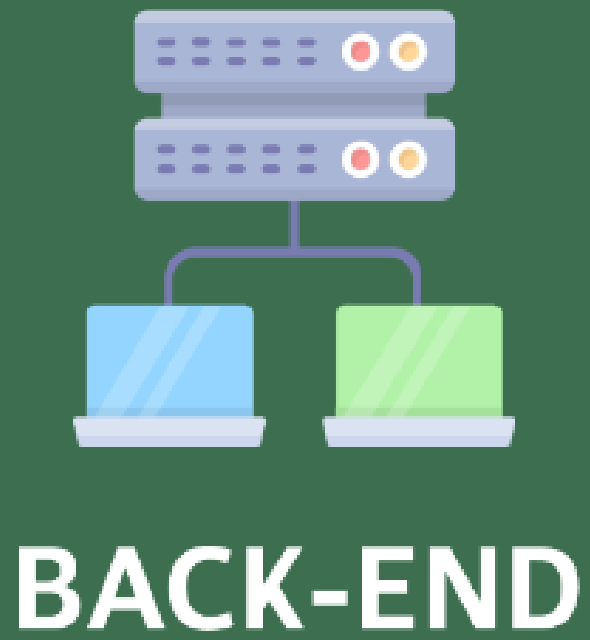


It's Your Life

with



자 이제



시작해볼까











서버는 결국

뭘 하죠?



Serve

+ er









요청한 거

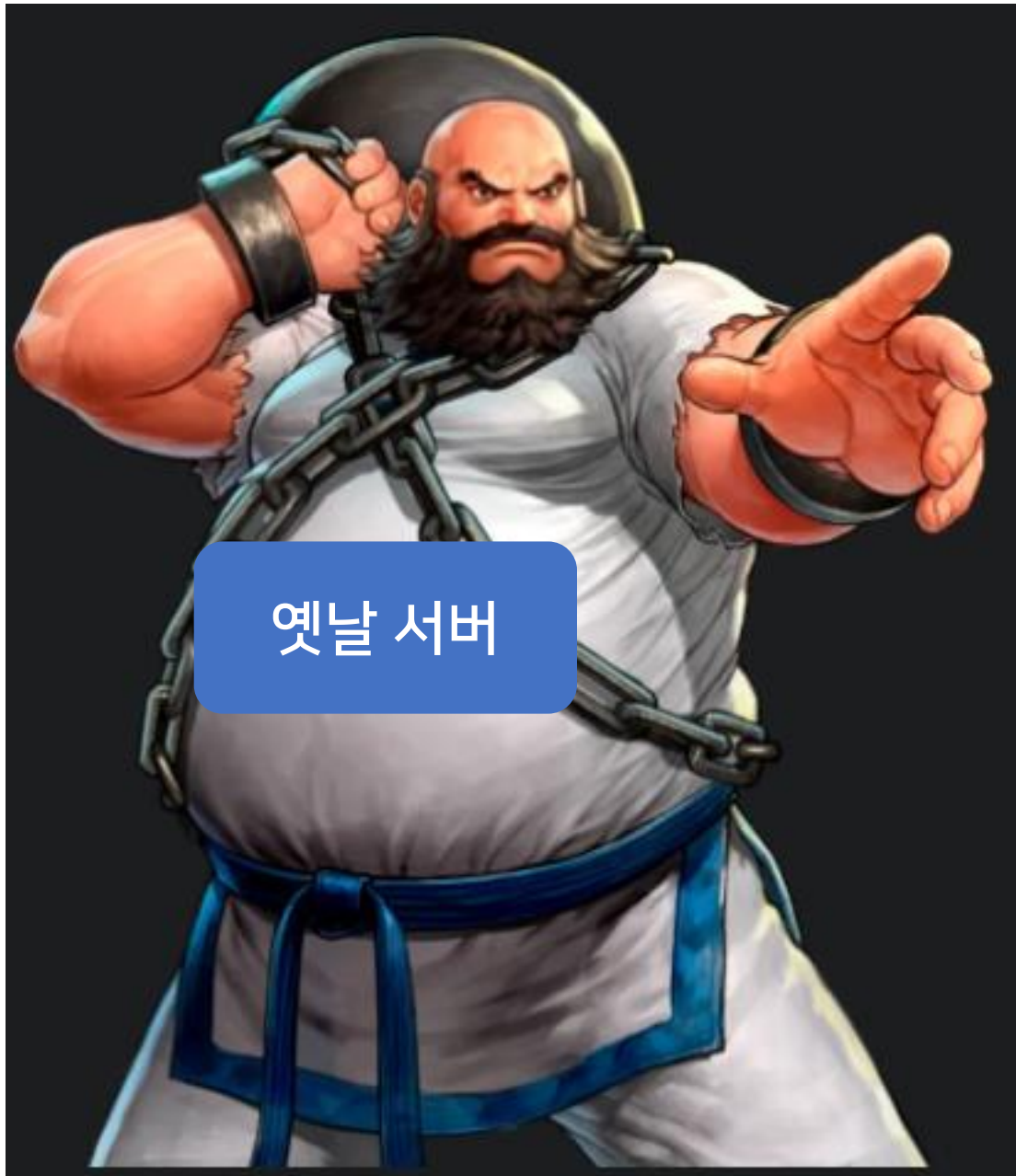




Servlet

Server + Applet





옛날 서버

무겁고 느리다









게재시 인터넷에 과려 처벌 규정에 이거 정보토시만번



Servlet



```
@WebServlet(name = "helloServlet", value = "/hello-servlet")
public class HelloServlet extends HttpServlet {
    private String message; 2 usages

    public void init() {
        System.out.println("init 호출");
        message = "Hello World!";
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>" + message + "</h1>");
        out.println("</body></html>");
    }
}
```

Tomcat







공짜입니다!











결국 서버가

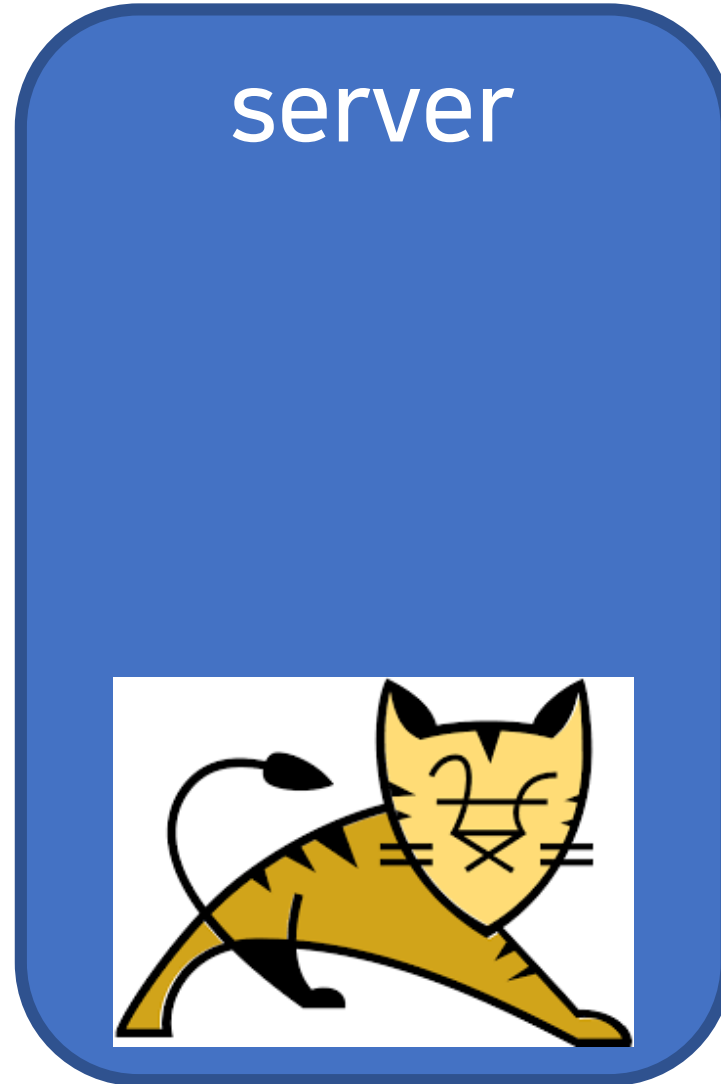
하는 일은!



주소 : <http://localhost:8080>



<http://localhost:8080>





주소 : <http://localhost:8080>

server

<http://localhost:8080>

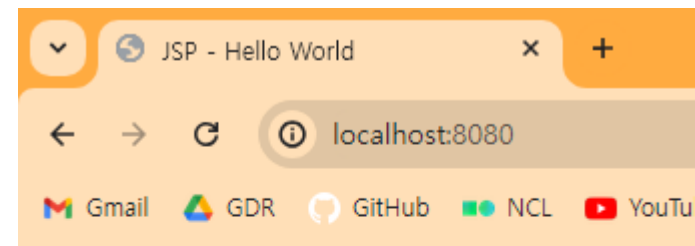


주소 : <http://localhost:8080>

server

index.jsp

<http://localhost:8080/>



Hello World!

[Hello Servlet](#)





기본 주소를 요청 하면
요 페이지를 가져다 줍니다!

```
src
├── main
│   ├── java
│   │   ├── org
│   │   │   ├── scoula
│   │   │   │   ├── dynamicweb
│   │   │   │   │   ├── HelloServlet
│   │   │   │   │   ├── LoopServlet
│   │   │   │   │   └── ResponseServlet
│   │   └── resources
│   └── webapp
│       ├── WEB-INF
│       │   ├── web.xml
│       │   ├── JSP hello.jsp
│       │   └── JSP index.jsp
```



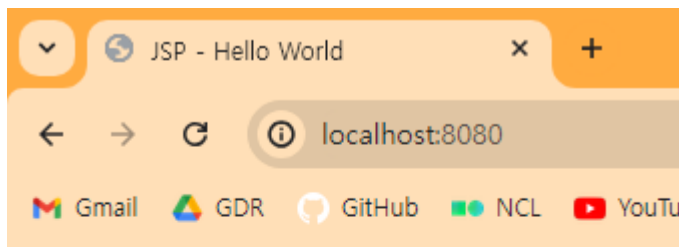
```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>JSP - Hello World</title>
</head>
<body>
<h1><%= "Hello World!" %>
</h1>
<br/>
<a href="hello-servlet">Hello Servlet</a>
</body>
</html>
```

HTML 처럼 생긴 걸
전달해 줍니다!

서버 입장에서는 HTML 은
의미가 없는 문자열일 뿐입니다



```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>JSP - Hello World</title>
</head>
<body>
<h1><%= "Hello World!" %>
</h1>
<br/>
<a href="hello-servlet">Hello
</body>
</html>
```



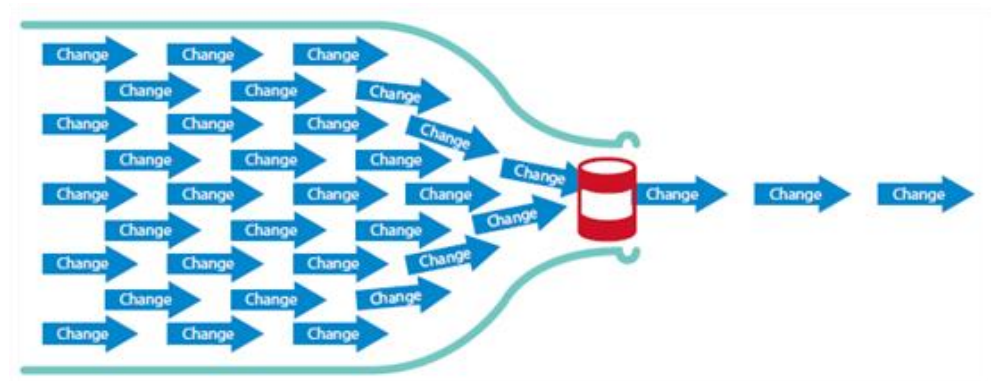
Hello World!

[Hello Servlet](#)

주소별로 업무를 나누자! 라우팅!



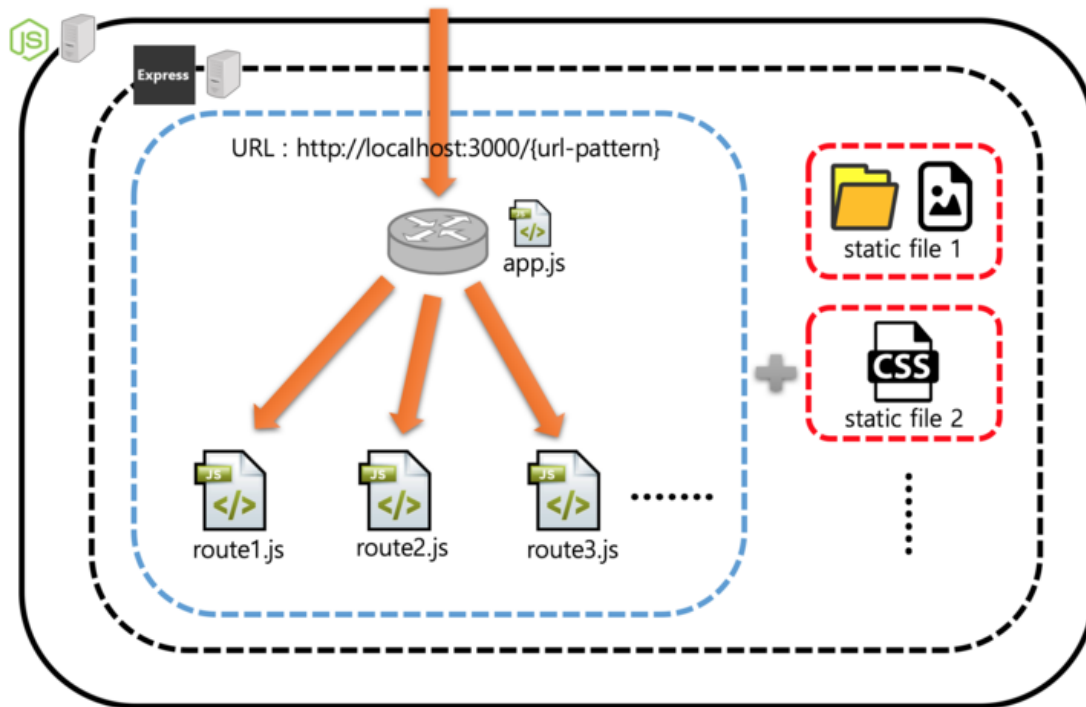
- 모든 요청을 하나의 주소에서 받으면 어떤 요청인지 구분이 어렵겠죠?
- 그리고 메인 서버 코드도 복잡하게 됩니다!



주소별로 업무를 나누자! 라우팅!



- 주소 요청에 따라 각각 담당하는 파일을 나눈다면!?
- 병목 현상 → 해결 / 코드 유지 보수 → 편해짐



© ResponseServlet

resources

webapp

WEB-INF

web.xml

JSP hello.jsp

JSP index.jsp

JSP loop.jsp

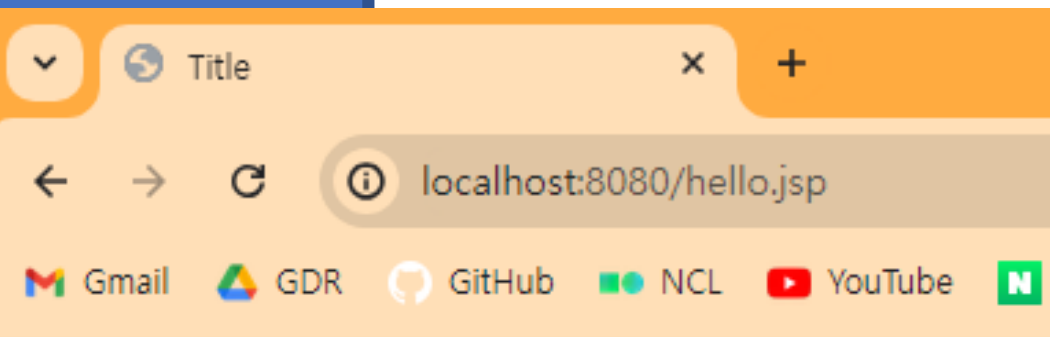
test

.gitignore

build.gradle

```
7  <!-- %>
8  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9  <html>
10 <head>
11     <title>Title</title>
12     <h1>안녕하세요!</h1>
13 </head>
14 <body>
15
16 </body>
17 </html>
18
```

<http://localhost:8080/hello.jsp>

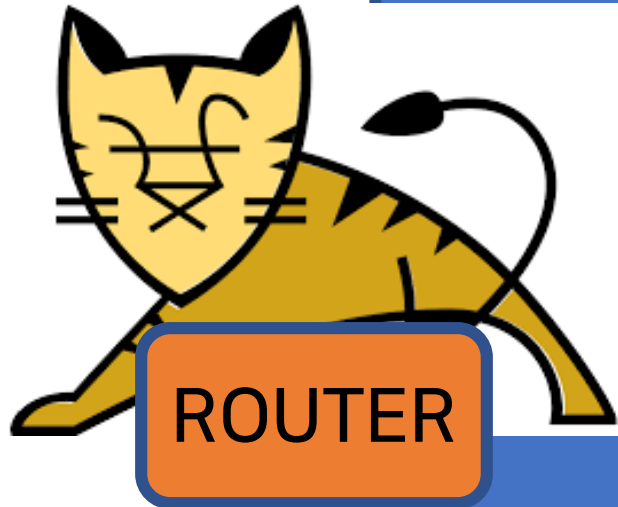


안녕하세요!

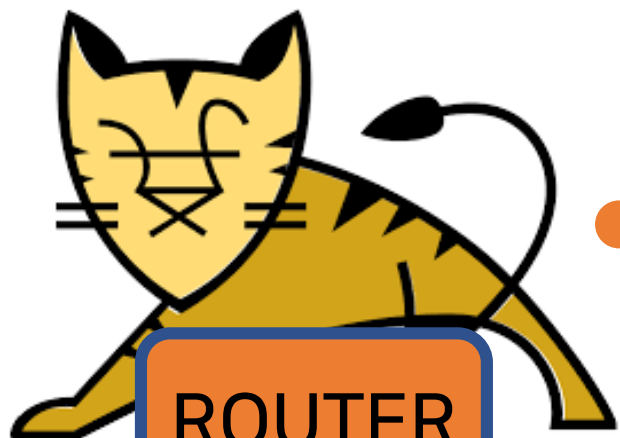
server

index.jsp

hello.jsp



http://localhost
:8080/



ROUTER

server

index.jsp

hello.jsp

<http://localhost:8080/hello.jsp>



ROUTER

server

index.jsp

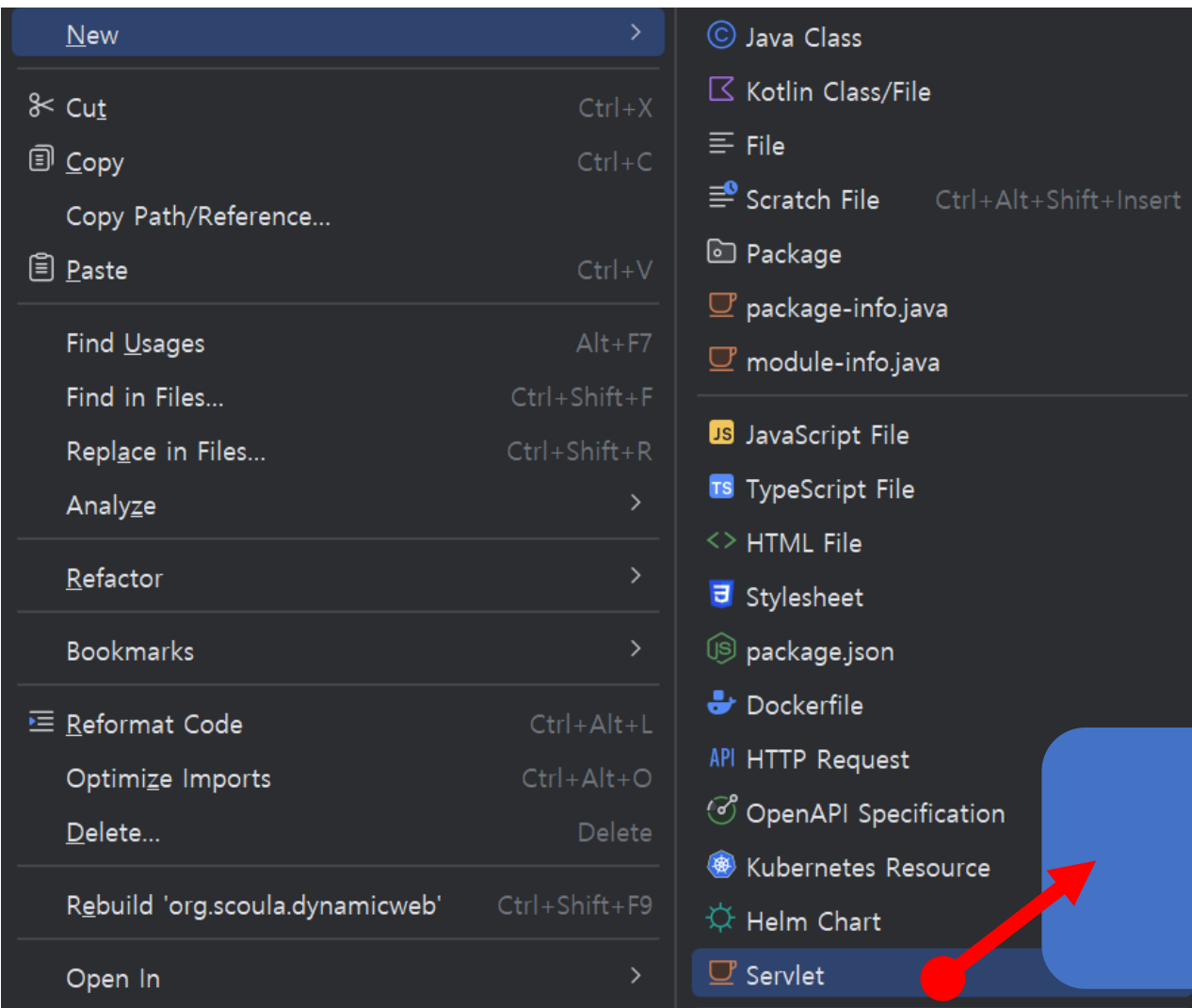
hello.jsp





Servlet

맛 보기!



Servlet 을 선택!

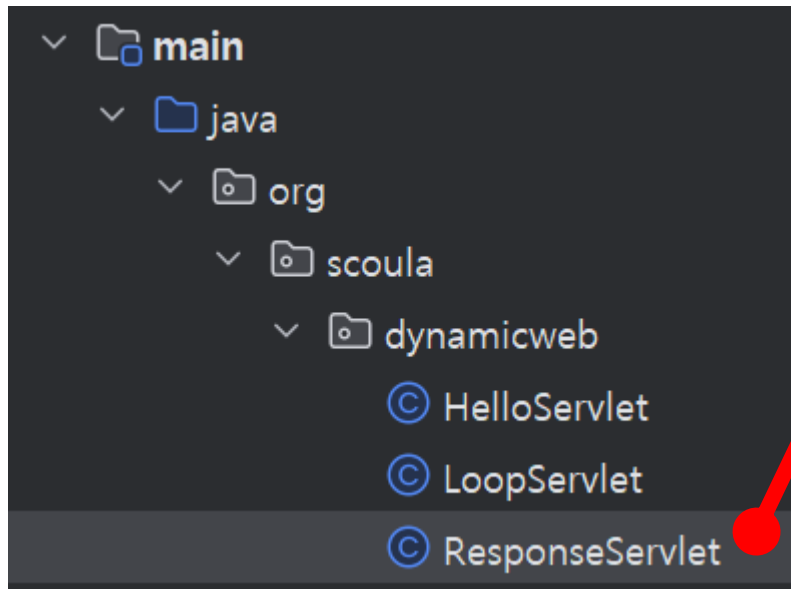
New Servlet

URL MAPPING:

ClassName:

OK Cancel

요청 주소 값과
클래스 이름을 지정!



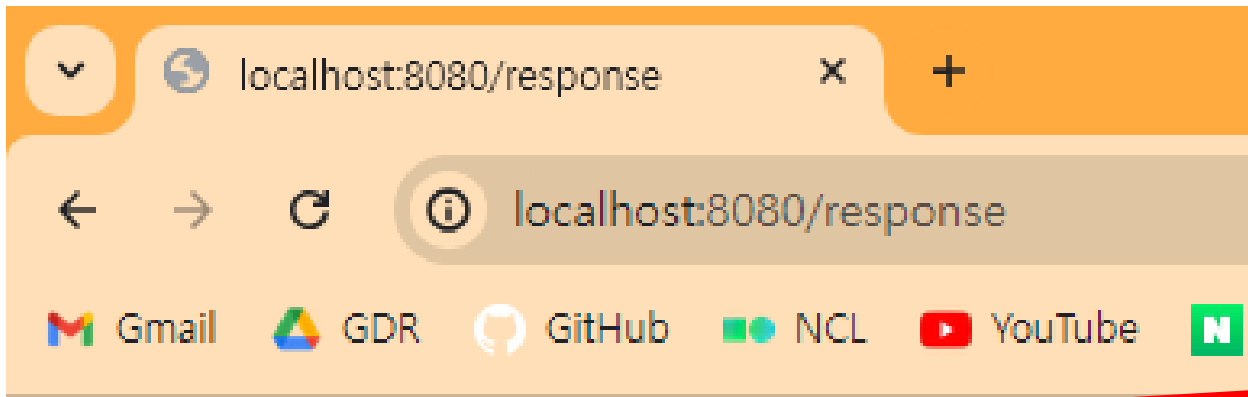
ResponseServlet 클래스가 추가!

```
@WebServlet("/response")
public class ResponseServlet extends HttpServlet {
    @Override no usages
    protected void doGet(HttpServletRequest request, HttpServletResponse
        response.setContentType("text/html;charset=utf-8");

    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("요청 성공");
    out.println("</body></html>");
}
```

/response 주소에 GET 방식으로
요청이 들어오면 해당 요청을
처리해주는 doGet 메서드!

간단한 HTML 페이지를
문자열로 만들어줘야 합니다!

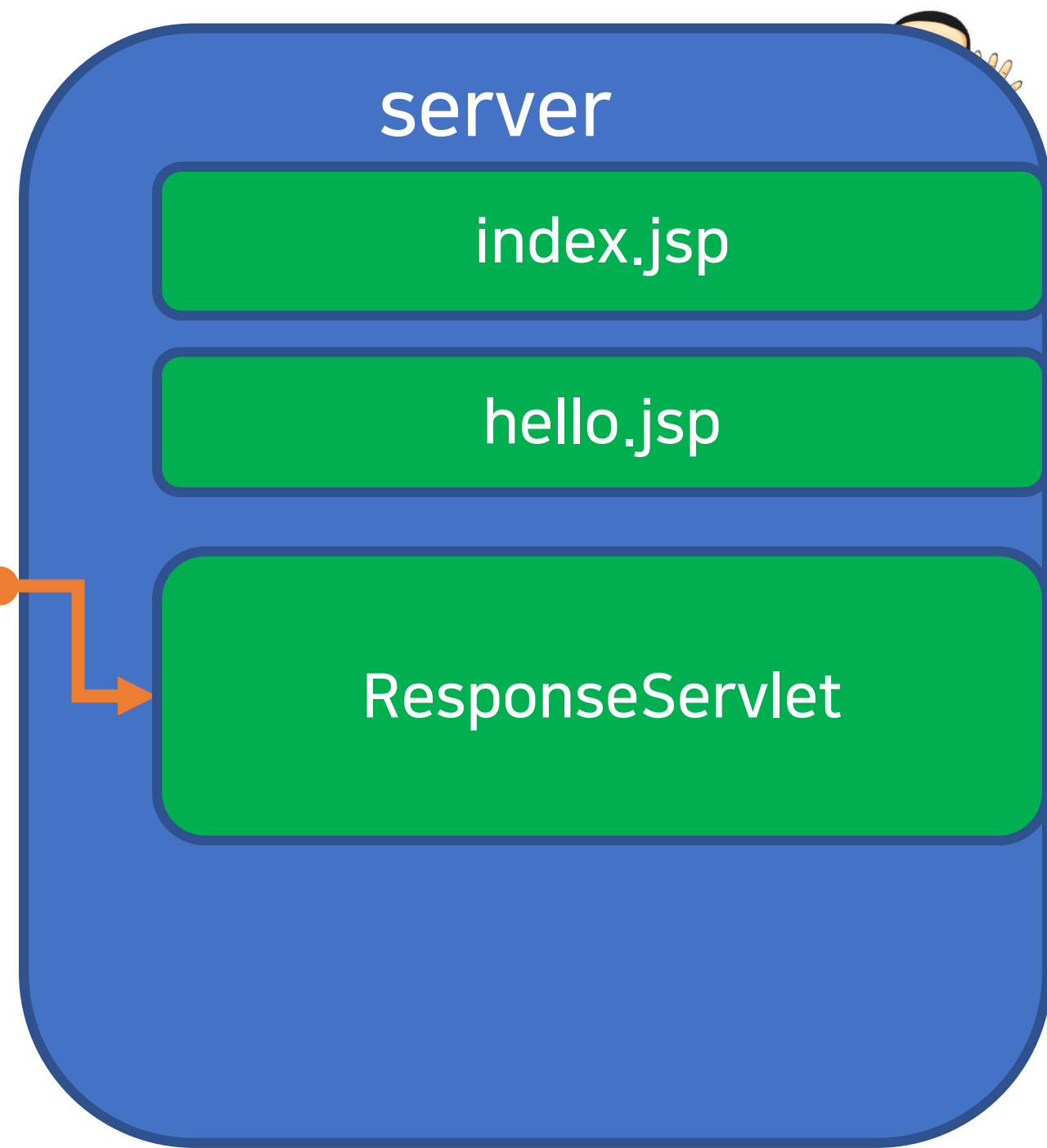
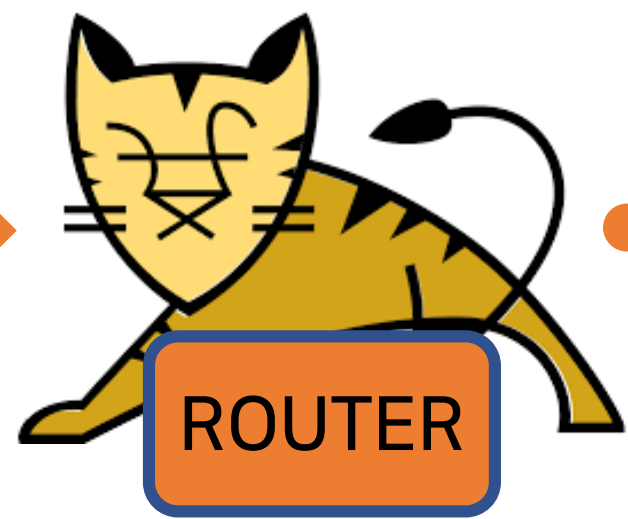


요청 성공

톰캣 서버에서 작성한
HTML 페이지가 출력 되는 것을 확인



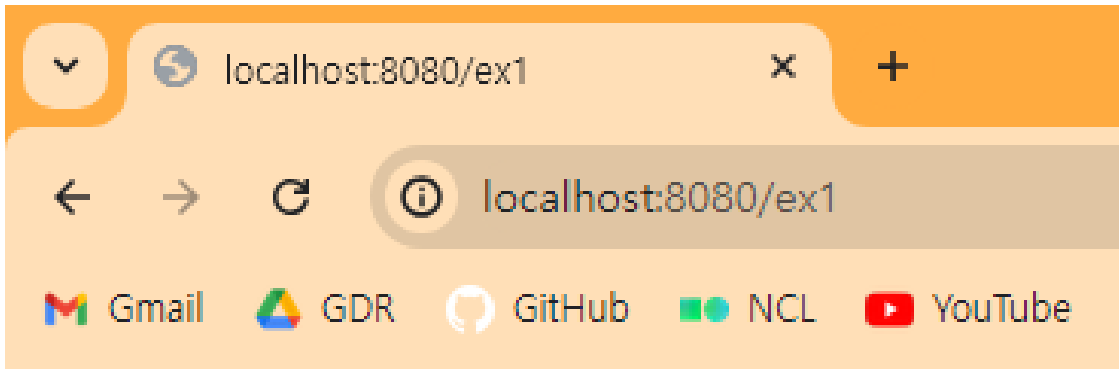
http://localhost:
8080/response



초간단 실습, 서블릿 페이지 작성하기



- /ex1 에 대응하는 서블릿 클래스를 작성해 주세요
- ex1 주소에 GET 방식으로 요청이 들어오면 <h1> 태그로 “오셨습니까!?”
를 출력하는 서블릿 클래스를 만들어 주세요!



오셨습니까!?

Servlet 의 단점!



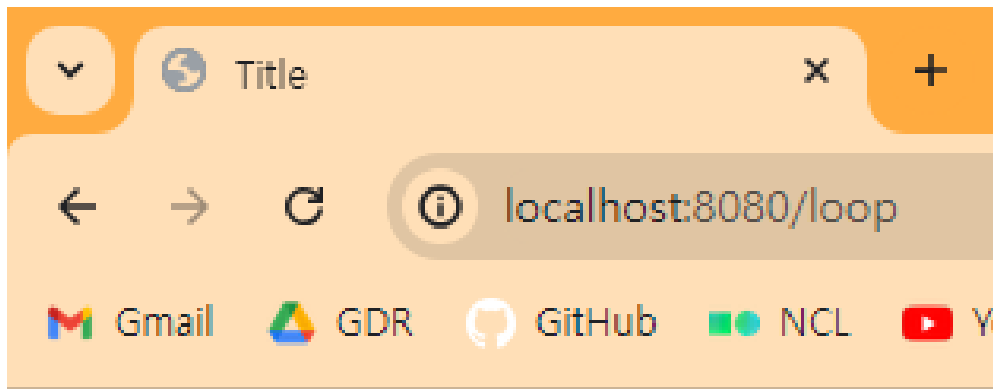




```
@WebServlet("/ex1")
public class ExServlet extends HttpServlet {
    @Override 1 usage
    protected void doGet(HttpServletRequest request, HttpServletResponse response) {
        response.setContentType("text/html;charset=utf-8");

        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>오셨습니까!?!</h1>");
        out.println("</body></html>");
    }
}
```

이런 방식으로 복잡한
웹페이지를 만들 수 있을까요!?



총 10 번 반복

- 1 번째 반복 중
- 2 번째 반복 중
- 3 번째 반복 중
- 4 번째 반복 중
- 5 번째 반복 중
- 6 번째 반복 중
- 7 번째 반복 중
- 8 번째 반복 중
- 9 번째 반복 중
- 10 번째 반복 중





```
public class LoopServlet extends HttpServlet {
    @Override 1 usage
    protected void doGet(HttpServletRequest request, HttpServletResponse response) {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        int max = 10;
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Title</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<p>총 " + max + " 번 반복</p>");
        out.println("<ul>");

        for (int i = 1; i <= max; i++) {
            out.println("<li>" + i + " 번째 반복 중</li>");
        }

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





이건 아니야!!



그래서 등장

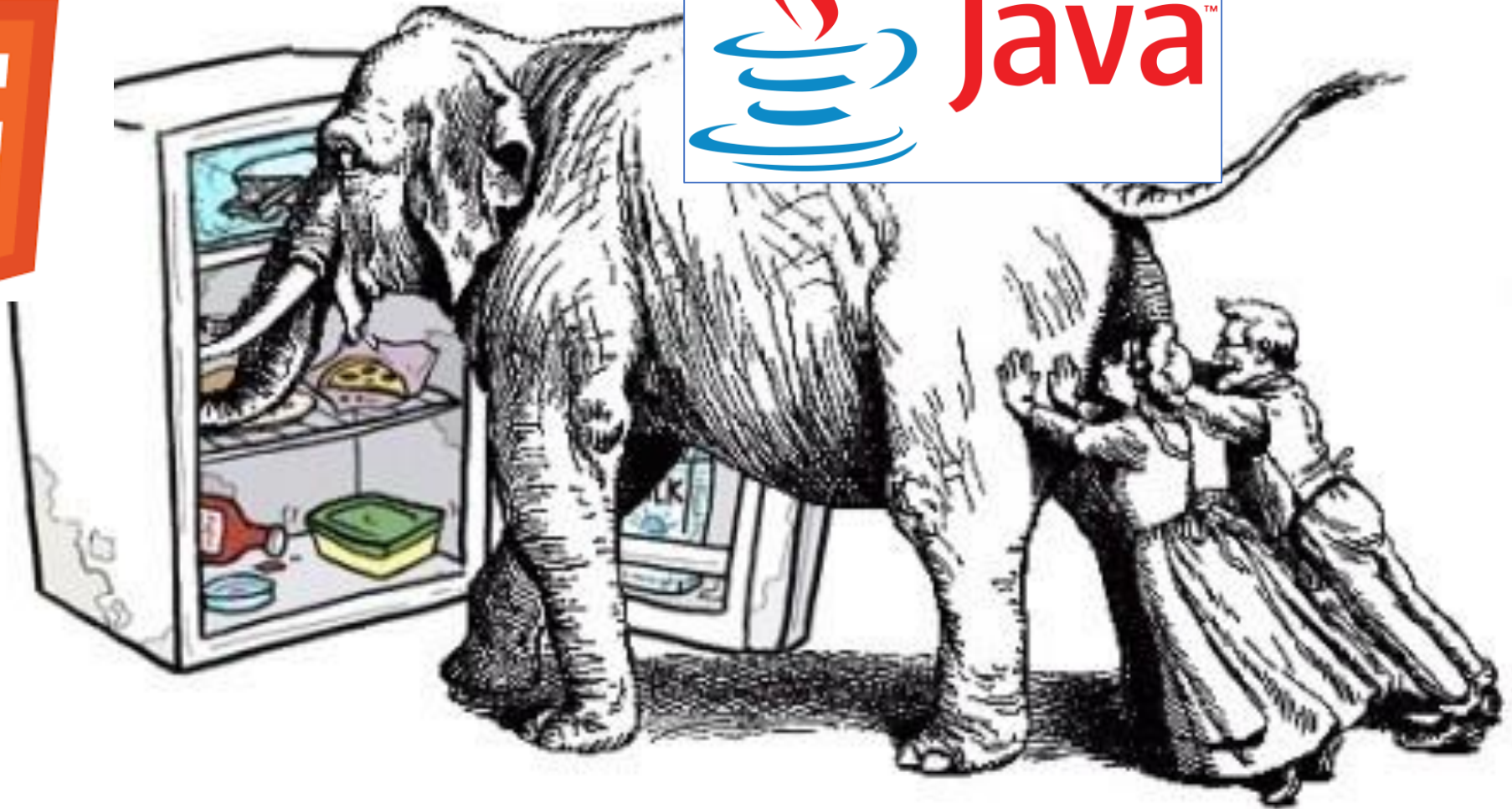
JSP는?

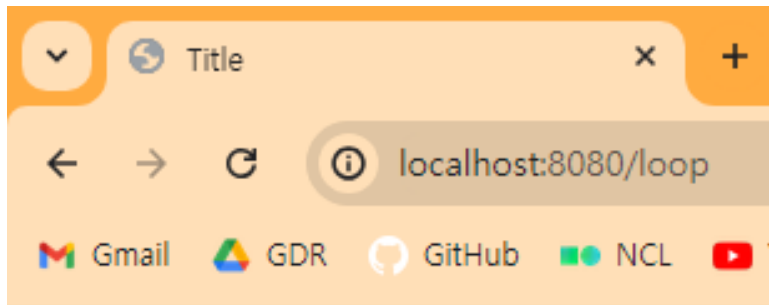


Java
Server

(HTML)
Page

パイナップーパーペン!
(Pineapple pen !)





총 10 번 반복

- 1 번째 반복 중
- 2 번째 반복 중
- 3 번째 반복 중
- 4 번째 반복 중
- 5 번째 반복 중
- 6 번째 반복 중
- 7 번째 반복 중
- 8 번째 반복 중
- 9 번째 반복 중
- 10 번째 반복 중

```
public class LoopServlet extends HttpServlet {  
    @Override 1 usage  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        response.setContentType("text/html;charset=UTF-8");  
        PrintWriter out = response.getWriter();  
  
        int max = 10;  
        out.println("<html>");  
        out.println("<head>");  
        out.println("<title>Title</title>");  
        out.println("</head>");  
        out.println("<body>");  
        out.println("<p>총 " + max + " 번 반복</p>");  
        out.println("<ul>");  
  
        for (int i = 1; i <= max; i++) {  
            out.println("<li>" + i + " 번째 반복 중</li>");  
        }  
  
        out.println("</ul>");  
        out.println("</body>");  
        out.println("</html>");  
    }  
}
```





```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <title>Title</title>
  <ul>
    <% int max = 10; %>
    <p>총 <%= max %> 번 반복</p>
    <ul>
      <%
        for (int i = 1; i <= max; i++) {
      %>
      <li><%= i %> 번째 반복 중</li>
      <%
        }
      %>
    </ul>
  </ul>
</head>
<body>
```

HTML 코드 내부에
JAVA 를 쓰고 싶으면 <% %> 를 쓰고
해당 블록 내부에 JAVA 코드를
쓰면 됩니다!!

변수를 바로 출력하고 싶으면!?
<%= %> 를 씁니다!!



```
<body>
  <div id="app">
    <ul>
      <p>총 {{ max }} 번 반복</p>
      <ul>
        <li v-for="i in max" :key="i">{{ i }} 번째 반복 중</li>
      </ul>
    </ul>
  </div>
```

Vue 를 쓰면 요렇게 구현이 됩니다!



JSP

주소 매칭

webapp
WEB-INF

web.xml

웹 서버 설정 파일을 열어 줍니다!





```
<servlet>
  <servlet-name>loop2</servlet-name>
  <jsp-file>/loop.jsp</jsp-file>
</servlet>
```

서블릿 이름과 jsp 파일 매칭

```
<servlet-mapping>
  <servlet-name>loop2</servlet-name>
  <url-pattern>/loop2</url-pattern>
</servlet-mapping>
```

서블릿 이름과 주소 매칭

Tomcat 9.0.91



설정이 바뀌면 서버를
재실행 해줘야 합니다!



Title

localhost:8080/loop2

Gmail GDR GitHub NCL YouTube

총 10 번 반복

- 1 번째 반복 중
- 2 번째 반복 중
- 3 번째 반복 중
- 4 번째 반복 중
- 5 번째 반복 중
- 6 번째 반복 중
- 7 번째 반복 중
- 8 번째 반복 중
- 9 번째 반복 중
- 10 번째 반복 중