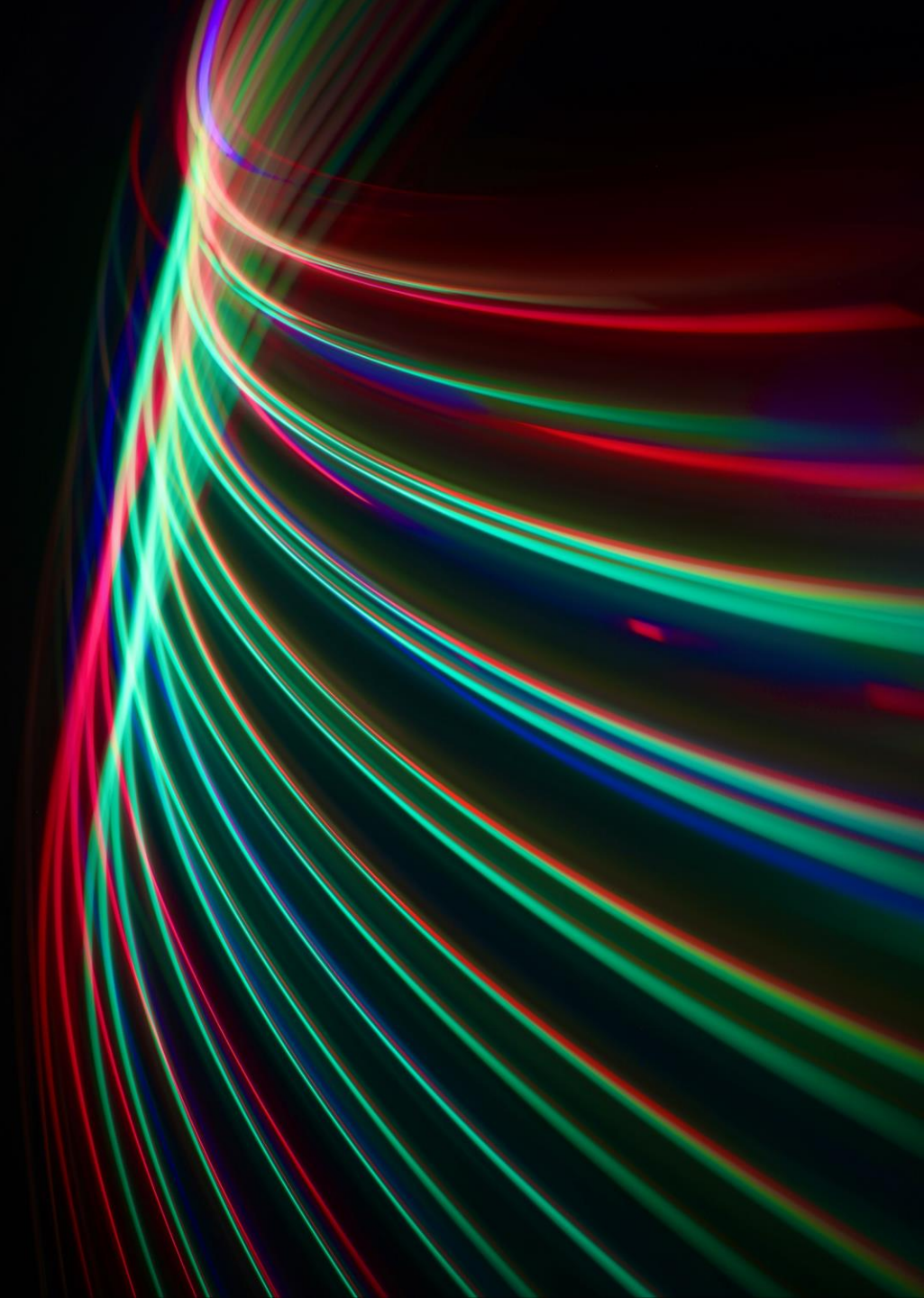


CGI, Servlet, JSP

Spring Web MVC 이해를 위한 첫 걸음

20210463 박연종

2024.07.10.



목차

Common Gateway Interface

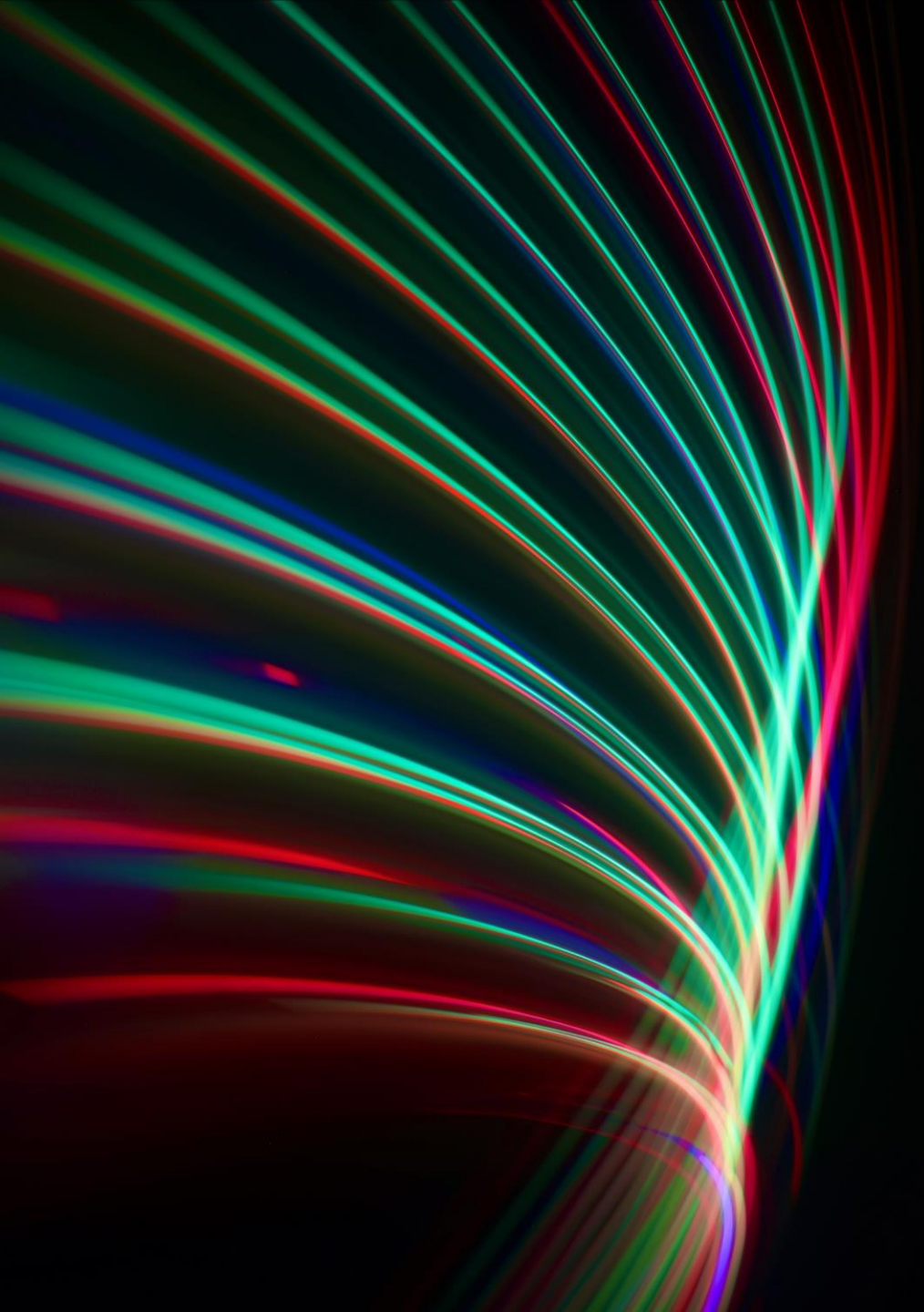
- ㄱ. CGI
- ㄴ. 확장 CGI

Servlet

- ㄱ. Servlet
- ㄴ. Servlet 동작 과정
- ㄷ. Servlet Container
- ㄹ. Servlet 생명 주기

JavaServer Pages (Jakarta Server Pages)

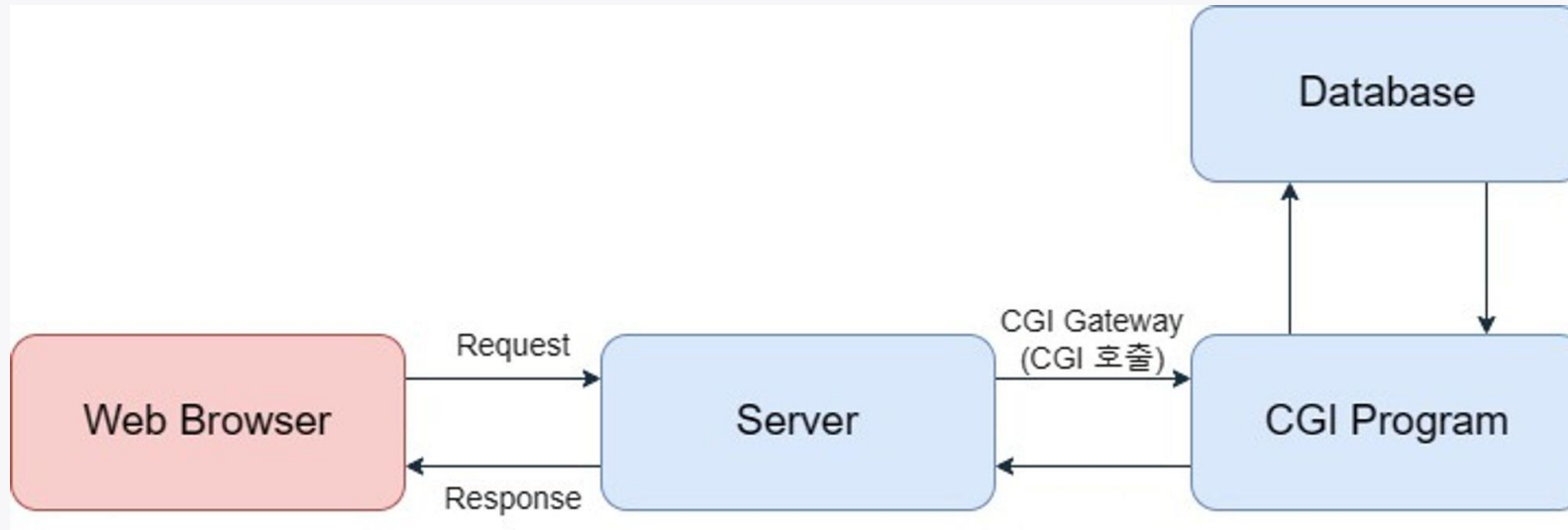
- ㄱ. JSP
- ㄴ. JSP 동작 과정



00.

Common Gateway Interface

Common Gateway Interface



- 웹 서버와 애플리케이션 간 통신을 위한 인터페이스
- 웹 서버가 직접 요청에 대한 프로그램을 호출하고 처리해 응답 결과를 브라우저에 전송



Common Gateway Interface

```
1 #!/usr/bin/perl
2
3 local ($buffer, @pairs, $pair, $name, $value, %FORM);
4 #Read in text
5 $ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;
6
7 if ($ENV{'REQUEST_METHOD'} eq "POST"){
8     read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
9 }else{
10     $buffer = $ENV{'QUERY_STRING'};
11 }
12
13 print "Content-type:text/html\r\n\r\n";
14 print "<html>";
15 print "<head>";
16 print "<title>CGI Post Method Example</title>";
17 print "</head>";
18 print "<body>";
19
20 # Split infomation into name/value pairs
21 @pairs = split(/&/, $buffer);
22 foreach $pair (@pairs){
23     ($name, $value) = split(/=/,$pair);
24     $value =~ tr/+// ;
25     $value =~ s/%(..)/pack("C",hex($1))/eg;
26     $FORM{$name} = $value;
27 }
28
29 $first_name = $FORM{'first_name'};
30 $last_name = $FORM{'last_name'};
31
32 print "<p>Hello, ${first_name} ${last_name}!</p>";
33 print "</body>";
34 print "</html>";
35
36 1;
```

POST localhost:81/cgi-bin/hello_post.pl Send

Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	first_name	lee			
<input checked="" type="checkbox"/>	last_name	jua			
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK 113 ms 339 B Save Response

Pretty Raw Preview Visualize

Hello, lee jua!

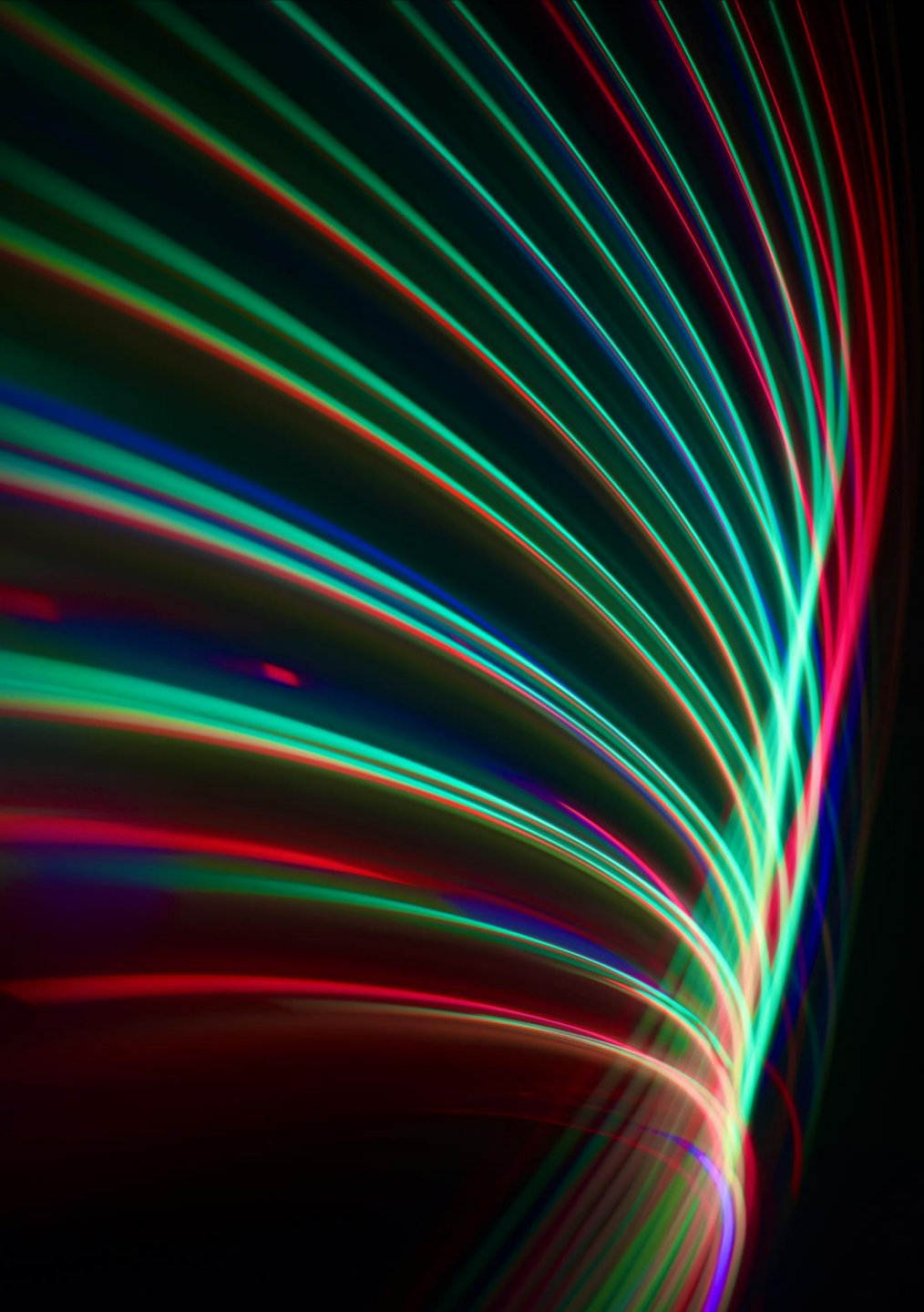
확장 CGI

CGI의 문제점

- 사용자 요청만큼 프로세스가 생성돼 느리고 메모리 낭비가 심함

확장 CGI

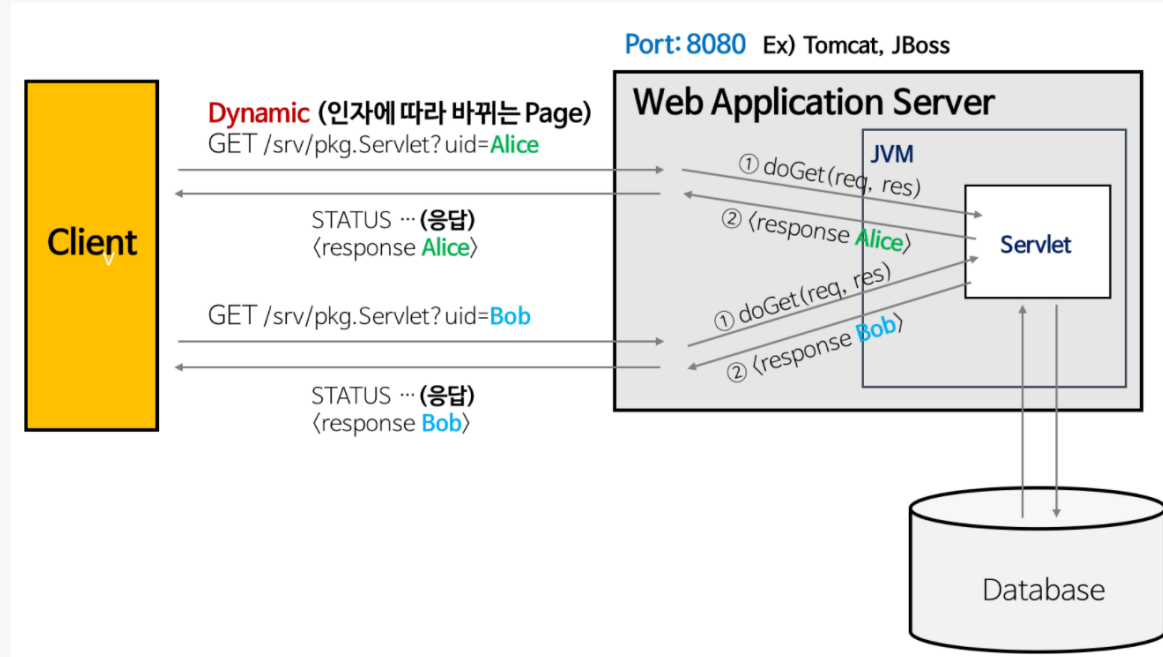
- 사용자 요청만큼 스레드를 생성해 기존 CGI보다 성능이 좋음
- ASP, PHP, Servlet 등이 확장 CGI에 해당



01.

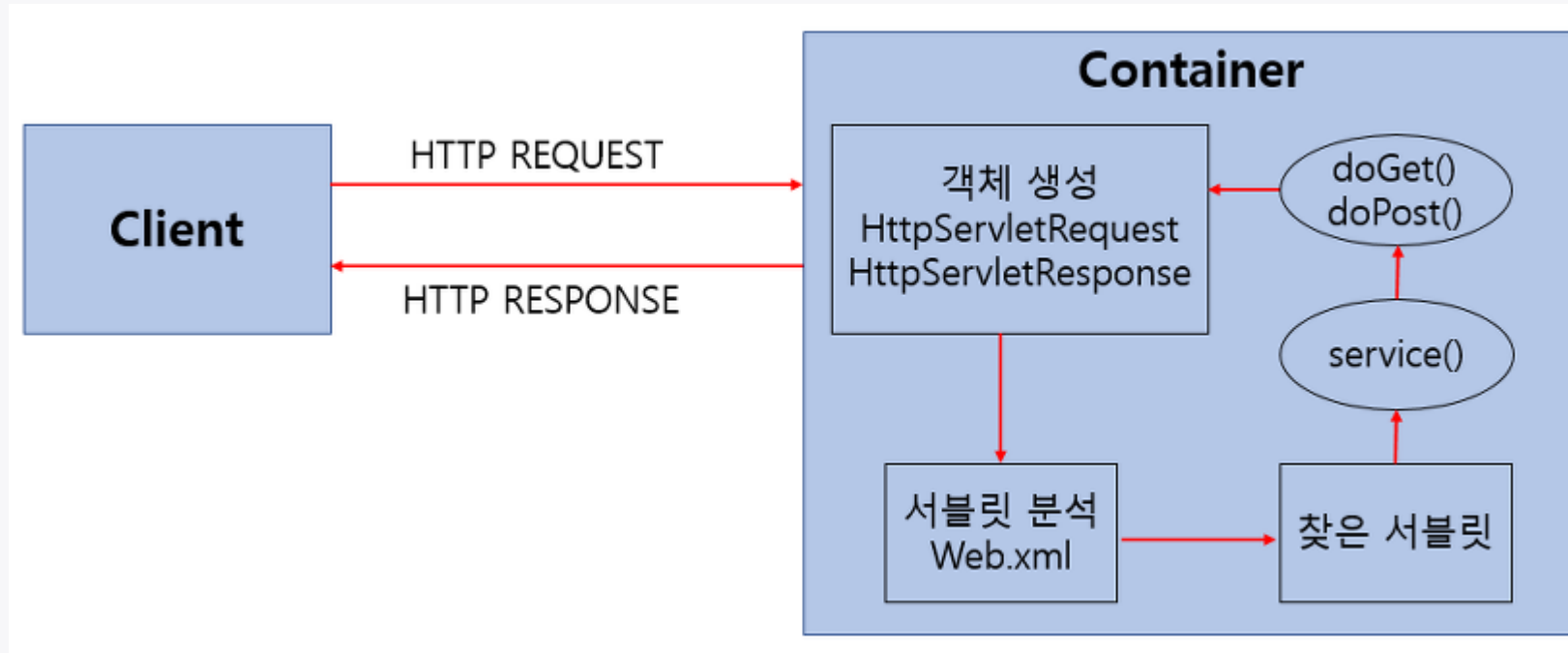
Servlet

Servlet



- Java 확장 CGI 프로그램
- HTTP 요청과 응답을 메서드 호출만으로 간단하고 체계적으로 다룰 수 있게 함

Servlet 동작 과정



Servlet Container

- Servlet을 관리해 주는 컨테이너
- 사용자 요청을 받고 응답할 수 있도록 웹 서버와 소켓으로 통신하는 프로그램

웹 서버와의 통신 지원

- 소켓 생성, listen, accept 등 일련의 반복적인 과정을 Servlet Container가 수행
- 개발자가 실질적으로 중요한 부분(*비즈니스 로직*)에 집중할 수 있게 함

- 서버 TCP/IP 대기, 소켓 연결
- HTTP 요청 메시지를 파싱해서 읽기
- POST 방식, /save URL 인지
- Content-Type 확인
- HTTP 메시지 바디 내용 파싱
 - username, age 데이터를 사용할 수 있게 파싱
- 저장 프로세스 실행
- 비즈니스 로직 실행
 - 데이터베이스에 저장 요청
- HTTP 응답 메시지 생성 시작
 - HTTP 시작 라인 생성
 - Header 생성
 - 메시지 바디에 HTML 생성에서 입력
- TCP/IP에 응답 전달, 소켓 종료

Servlet Container

- Servlet을 관리해 주는 컨테이너
- 사용자 요청을 받고 응답할 수 있도록 웹 서버와 소켓으로 통신하는 프로그램

웹 서버와의 통신 지원

- 소켓 생성, listen, accept 등 일련의 반복적인 과정을 Servlet Container가 수행
- 개발자가 실질적으로 중요한 부분(*비즈니스 로직*)에 집중할 수 있게 함

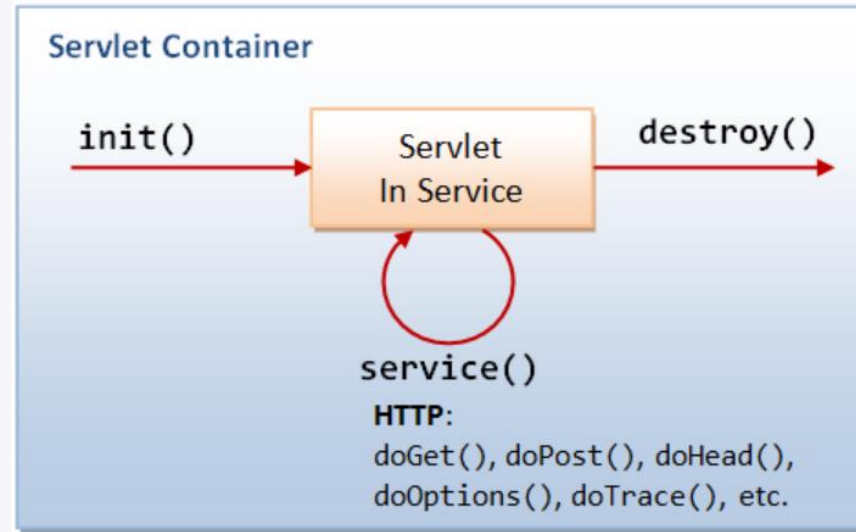
Servlet 생명 주기 관리

- Servlet 클래스를 인스턴스화하고 HTTP 요청과 응답 발생 시 적절한 메서드 호출

멀티스레드 지원 및 관리

- HTTP 요청마다 새로운 스레드를 생성하고 HTTP 응답 후 스레드가 자동으로 소멸
- 개발자가 스레드 관리에 신경 쓰지 않아도 됨

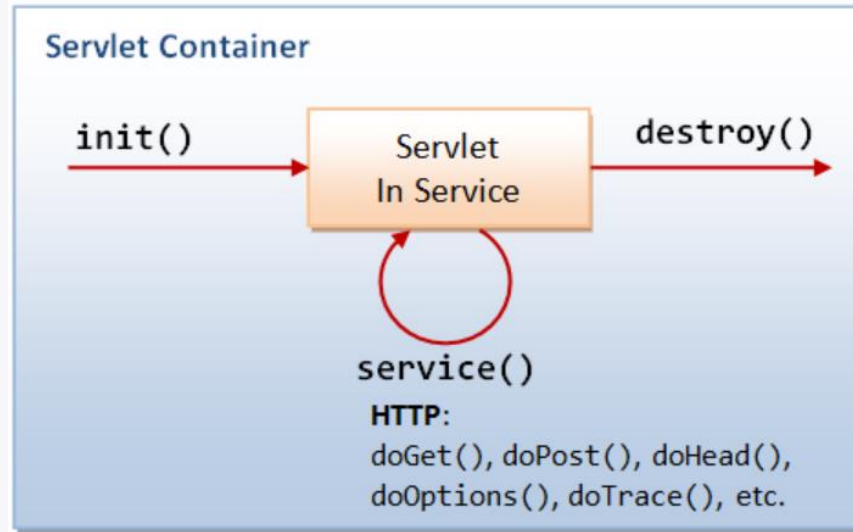
Servlet 생명 주기



init()

- HTTP 요청이 들어오면 Servlet Container는 해당 Servlet이 메모리 상에 적재돼 있는지 확인
- Servlet 인스턴스가 없으면 init() 메서드를 호출해 메모리 상에 적재
- Servlet 인스턴스가 있으면 해당 Servlet의 service() 메서드를 호출

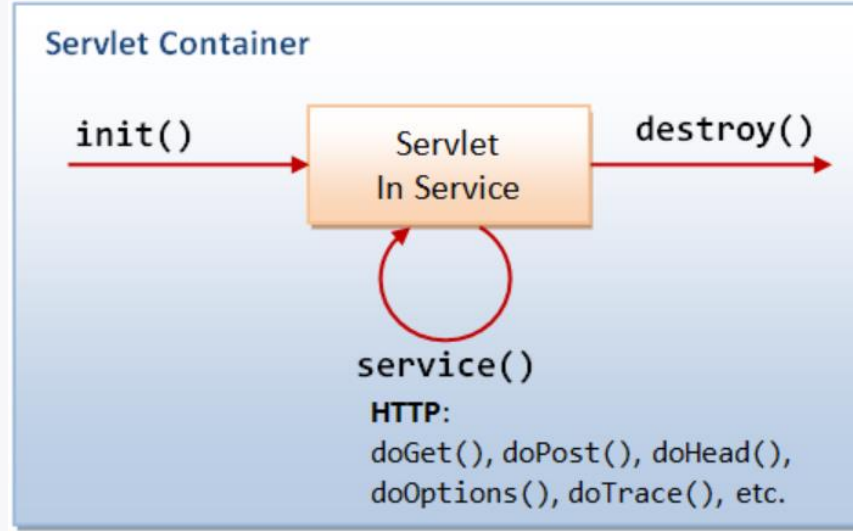
Servlet 생명 주기



주요 메서드

- HTTP 요청마다 생성된 `HttpServletRequest`, `HttpServletResponse` 객체를 전달 받음
- HTTP 요청 메서드를 확인해 해당하는 HTTP 요청 메서드의 메서드를 실행함
- 개발자는 해당 메서드만을 오버라이딩해 사용함으로써 실질적으로 중요한 부분(비즈니스 로직)에 집중할 수 있음

Servlet 생명 주기



종료 메서드

- Servlet Container가 Servlet에 종료 요청 시 실행되는 메서드



02.

Servlet 사용해 보기

Servlet 사용해 보기

```
12 @WebServlet(name = "memberFormServlet", urlPatterns = {"/servlet/members/new-form"})
13 public class MemberFormServlet extends HttpServlet {
14     ...
15     private final MemberRepository memberRepository = MemberRepository.getInstance(); no usages
16     ...
17     @Override
18     @protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19         response.setContentType("text/html");
20         response.setCharacterEncoding("utf-8");
21         ...
22         PrintWriter w = response.getWriter();
23         ...
24         w.write("<!DOCTYPE html>\n" +
25             "<html>\n" +
26             "<head>\n" +
27             "    <meta charset=\"UTF-8\">\n" +
28             "    <title>Title</title>\n" +
29             "</head>\n" +
30             "<body>\n" +
31             "    <form action=\"/servlet/members/save\" method=\"post\">\n" +
32             "        username: <input type=\"text\" name=\"username\" />\n" +
33             "        age: <input type=\"text\" name=\"age\" />\n" +
34             "        <button type=\"submit\">전송</button>\n" +
35             "    </form>\n" +
36             "</body>\n" +
37             "</html>\n"
38         );
39     }
40 }
```

username: age:

Servlet 사용해 보기

```
13 @WebServlet(name = "memberSaveServlet", urlPatterns = "/servlet/members/save")
14 public class MemberSaveServlet extends HttpServlet {
15     private final MemberRepository memberRepository = MemberRepository.getInstance(); // usage
16     ...
17     ... @Override
18     @@@ protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19         System.out.println("MemberSaveServlet.service");
20         String username = request.getParameter("username");
21         int age = Integer.parseInt(request.getParameter("age"));
22         ...
23         Member member = new Member(username, age);
24         memberRepository.save(member);
25         ...
26         response.setContentType("text/html");
27         response.setCharacterEncoding("utf-8");
28         ...
29         PrintWriter w = response.getWriter();
30         w.write("<html>\n" +
31             "<head>\n" +
32             "    <meta charset='UTF-8'>\n" +
33             "</head>\n" +
34             "<body>\n" +
35             "    성공\n" +
36             "    <ul>\n" +
37             "        <li>id="+member.getId()+"</li>\n" +
38             "        <li>username="+member.getUsername()+"</li>\n" +
39             "        <li>age="+member.getAge()+"</li>\n" +
40             "    </ul>\n" +
41             "    <a href='\"/index.html\">메인</a>\n" +
42             "</body>\n" +
43             "</html>"
44         );
45     }
46 }
```

성공

- id=1
- username=PARK, Yeonjong
- age=21

메인

Servlet 사용해 보기

```
14 @WebServlet(name = "memberListServlet", urlPatterns = "/servlet/members")
15 public class MemberListServlet extends HttpServlet {
16     private final MemberRepository memberRepository = MemberRepository.getInstance(); // 1 usage
17     // ...
18     @Override
19     @@@ protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
20         List<Member> members = memberRepository.findAll();
21         // ...
22         response.setContentType("text/html");
23         response.setCharacterEncoding("utf-8");
24         // ...
25         PrintWriter w = response.getWriter();
26         w.write(s: "<html>\n");
27         w.write(s: "<head>\n");
28         w.write(s: "<meta charset='UTF-8'>\n");
29         w.write(s: "<title>Title</title>\n");
30         w.write(s: "</head>\n");
31         w.write(s: "<body>\n");
32         w.write(s: "<a href='\"/index.html\"'>메인</a>\n");
33         w.write(s: "<table>\n");
34         w.write(s: "<thead>\n");
35         w.write(s: "<th>id</th>\n");
36         w.write(s: "<th>username</th>\n");
37         w.write(s: "<th>age</th>\n");
38         w.write(s: "</thead>\n");
39         w.write(s: "<tbody>\n");
40         for (Member member : members) {
41             w.write(s: "<tr>\n");
42             w.write(s: "<td>" + member.getId() + "</td>\n");
43             w.write(s: "<td>" + member.getUsername() + "</td>\n");
44             w.write(s: "<td>" + member.getAge() + "</td>\n");
45             w.write(s: "</tr>\n");
46         }
47         w.write(s: "</tbody>\n");
48         w.write(s: "</table>\n");
49         w.write(s: "</body>\n");
50         w.write(s: "</html>\n");
51     }
52 }
```

메인

id	username	age
1	PARK, Yeonjong	21
2	Ryu, Sangjin	21
3	Yoon, Huidong	21
4	Kwon, Dawoon	22
5	Jeong, Yeongun	22
6	Park, Seongjin	21
7	Shin, Dongyeon	22
8	Woon, Dongseon	22
9	Kim, Donghyeon	22
10	Kim, juhyeon	24
11	Kim, Seongmin	22
12	O, Seungmin	21
13	Eom, Huijae	21
14	Song, Chanhyeok	20
15	Kim, Seongmin	21



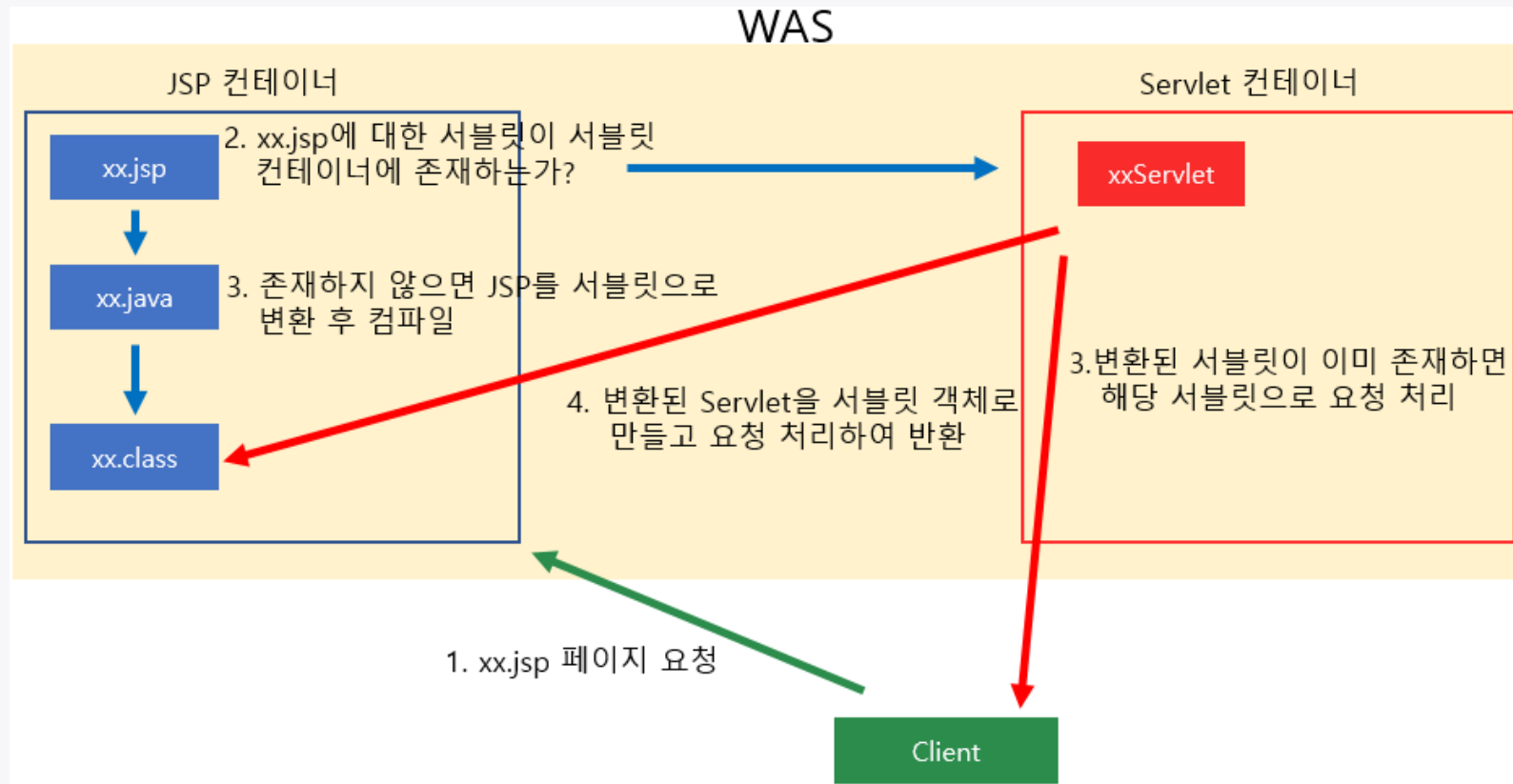
03.

JavaServer Pages

(Jakarta Server Pages)

JavaServer Pages (Jakarta Server Pages)

- 동적 웹 페이지를 만드는 서버 사이드 스크립트 언어





04.

JSP 사용해 보기

JSP 사용해 보기 (1)

```
1  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2  <html>
3  <head>
4  .... <title>Title</title>
5  </head>
6  <body>
7  .... <form action="/jsp/members/save.jsp" method="post">
8  ....     username: <input type="text" name="username" />
9  ....     age: ..... <input type="text" name="age" />
10 ....     <button type="submit">전송</button>
11 .... </form>
12 </body>
13 </html>
```

username: age:

JSP 사용해 보기 (1)

```
1  <%@ page import="com.patulus.helloServlet.domain.member.Member" %>
2  <%@ page import="com.patulus.helloServlet.domain.member.MemberRepository" %>
3  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
4  <%
5      // request, response는 JSP가 문법상 지원
6      MemberRepository memberRepository = MemberRepository.getInstance();
7
8      System.out.println("MemberSaveServlet.service");
9      String username = request.getParameter("username");
10     int age = Integer.parseInt(request.getParameter("age"));
11
12     Member member = new Member(username, age);
13     memberRepository.save(member);
14 %>
15 <html>
16 <head>
17     <title>Title</title>
18 </head>
19 <body>
20     성공
21     <ul>
22         <li>id=<%= member.getId() %></li>
23         <li>username=<%= member.getUsername() %></li>
24         <li>age=<%= member.getAge() %></li>
25     </ul>
26     <a href="/index.html">메인</a>
27 </body>
28 </html>
```

성공

- id=1
- username=PARK, Yeonjong
- age=21

메인

JSP 사용해 보기 (1)

```
1  <%@ page import="com.patulus.helloServlet.domain.member.MemberRepository" %>
2  <%@ page import="com.patulus.helloServlet.domain.member.Member" %>
3  <%@ page import="java.util.List" %>
4  <%@ page contentType="text/html; charset=UTF-8" language="java" %>
5  <%
6      MemberRepository memberRepository = MemberRepository.getInstance();
7      List<Member> members = memberRepository.findAll();
8  %>
9  <html>
10     <head>
11         <title>Title</title>
12     </head>
13     <body>
14         <a href="/index.html">메인</a>
15         <table>
16             <thead>
17                 <th>id</th>
18                 <th>username</th>
19                 <th>age</th>
20             </thead>
21             <tbody>
22                 <%
23                     for (Member member : members) {
24                         out.write("<tr>\n");
25                         out.write("<td>" + member.getId() + "</td>\n");
26                         out.write("<td>" + member.getUsername() + "</td>\n");
27                         out.write("<td>" + member.getAge() + "</td>\n");
28                         out.write("<tr>\n");
29                     }
30                 %>
31             </tbody>
32         </table>
33     </body>
34 </html>
```

메인

id	username	age
1	PARK, Yeonjong	21
2	Ryu, Sangjin	21
3	Yoon, Huidong	21
4	Kwon, Dawoon	22
5	Jeong, Yeongun	22
6	Park, Seongjin	21
7	Shin, Dongyeon	22
8	Woon, Dongseon	22
9	Kim, Donghyeon	22
10	Kim, Juhyeon	24
11	Kim, Seongmin	22
12	O, Seungmin	21
13	Eom, Huijae	21
14	Song, Chanhyeok	20
15	Kim, Seongmin	21

JSP 사용해 보기 (2)

```
11 @WebServlet(name = "mvcMemberFormServlet", urlPatterns = "/servlet-mvc/members/new-form")
12 public class MvcMemberFormServlet extends HttpServlet {
13     ...
14     ...@Override
15     ...protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
16         ...String viewPath = "/WEB-INF/views/new-form.jsp";
17         ...RequestDispatcher dispatcher = request.getRequestDispatcher(viewPath);
18         ...dispatcher.forward(request, response); // JSP를 찾아 호출
19     }
20 }
```

```
1 <%@ page contentType="text/html; charset=UTF-8" language="java"%>
2 <html>
3 <head>
4     ...<meta charset="UTF-8">
5     ...<title>Title</title>
6 </head>
7 <body>
8     <!-- 상대경로 사용, [현재 URL이 속한 계층 경로 + /save] -->
9     <form action="/save" method="post">
10         ...username: <input type="text" name="username" />
11         ...age: ...<input type="text" name="age" />
12         ...<button type="submit">전송</button>
13     </form>
14 </body>
15 </html>
```

username: age:

JSP 사용해 보기 (2)

```
14 @WebServlet(name = "mvcMemberListServlet", urlPatterns = "/servlet-mvc/members")
15 public class MvcMemberListServlet extends HttpServlet {
16     ...
17     private final MemberRepository memberRepository = MemberRepository.getInstance(); 1 usage
18     ...
19     @Override
20     @@@@ protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
21         List<Member> members = memberRepository.findAll();
22         ...
23         // Model에 담는다
24         request.setAttribute("members", members);
25         ...
26         // View에 넘긴다
27         String viewPath = "/WEB-INF/views/members.jsp";
28         RequestDispatcher dispatcher = request.getRequestDispatcher(viewPath);
29         dispatcher.forward(request, response);
30     }
31 }
```

```
1 <%@ page import="com.patulus.helloServlet.domain.member.Member" %>
2 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
3 <html>
4     <head>
5         <title>Title</title>
6     </head>
7     <body>
8         ... 성공
9         <ul>
10             <li>id=<%= ((Member)request.getAttribute("member")).getId() %></li>-->
11             <li>username=<%= ((Member)request.getAttribute("member")).getUsername() %></li>-->
12             <li>age=<%= ((Member)request.getAttribute("member")).getAge() %></li>-->
13             <li>id=${member.id}</li>
14             <li>username=${member.username}</li>
15             <li>age=${member.age}</li>
16         </ul>
17         <a href="/index.html">메인</a>
18     </body>
19 </html>
```

성공

- id=1
- username=PARK, Yeonjong
- age=21

메인

JSP 사용해 보기 (2)

```
14 @WebServlet(name = "mvcMemberListServlet", urlPatterns = "/servlet-mvc/members")
15 public class MvcMemberListServlet extends HttpServlet {
16     ...
17     private final MemberRepository memberRepository = MemberRepository.getInstance(); 1 usage
18     ...
19     @Override
20     @@@ protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
21         List<Member> members = memberRepository.findAll();
22         ...
23         // Model에 담는다
24         request.setAttribute("members", members);
25         ...
26         // View에 넘긴다
27         String viewPath = "/WEB-INF/views/members.jsp";
28         RequestDispatcher dispatcher = request.getRequestDispatcher(viewPath);
29         dispatcher.forward(request, response);
30     }
31 }
```

```
1 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
3 <html>
4 <head>
5     <title>Title</title>
6 </head>
7 <body>
8     <a href="/index.html">메인</a>
9     <table>
10         <thead>
11             <th>id</th>
12             <th>username</th>
13             <th>age</th>
14         </thead>
15         <tbody>
16             <c:forEach var="item" items="{members}">
17                 <tr>
18                     <td>${item.id}</td>
19                     <td>${item.username}</td>
20                     <td>${item.age}</td>
21                 </tr>
22             </c:forEach>
23         </tbody>
24     </table>
25 </body>
26 </html>
```

메인

id	username	age
1	PARK, Yeonjong	21
2	Ryu, Sangjin	21
3	Yoon, Huidong	21
4	Kwon, Dawoon	22
5	Jeong, Yeongun	22
6	Park, Seongjin	21
7	Shin, Dongyeon	22
8	Woon, Dongseon	22
9	Kim, Donghyeon	22
10	Kim, Juhyeon	24
11	Kim, Seongmin	22
12	O, Seungmin	21
13	Eom, Huijae	21
14	Song, Chanhyeok	20
15	Kim, Seongmin	21

감사합니다

Email / park@duck.com

Insta / [@yeonjong.park](https://www.instagram.com/yeonjong.park)

GitHub / [patulus](https://github.com/patulus)

