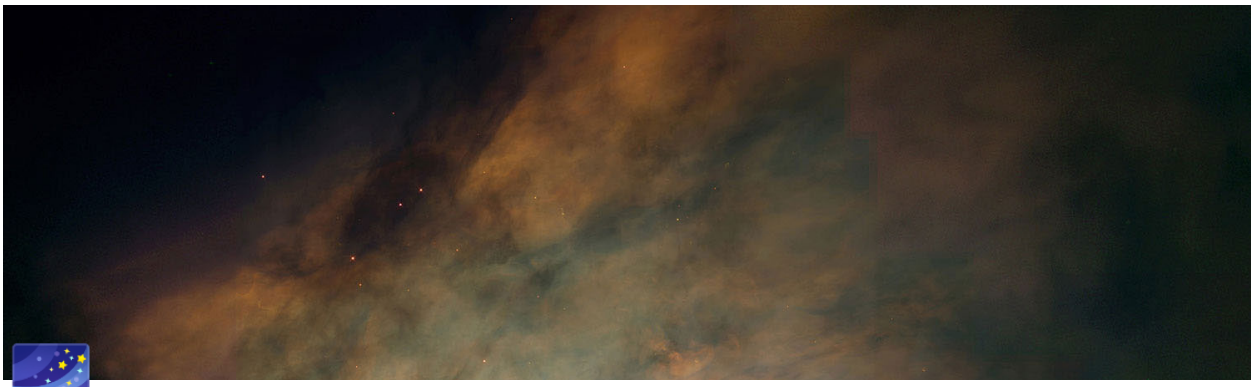


Web Technology Overview

WEB 이란

월드 와이드 웹

138개 언어

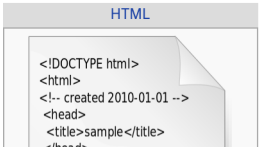
위키백과, 우리 모두의 백과사전.

이 문서는 인터넷의 정보 공간에 대해 설명하고 있습니다. 다른 뜻에 대해서는 웹, WWW (동음이의) 문서를 참고하십시오.

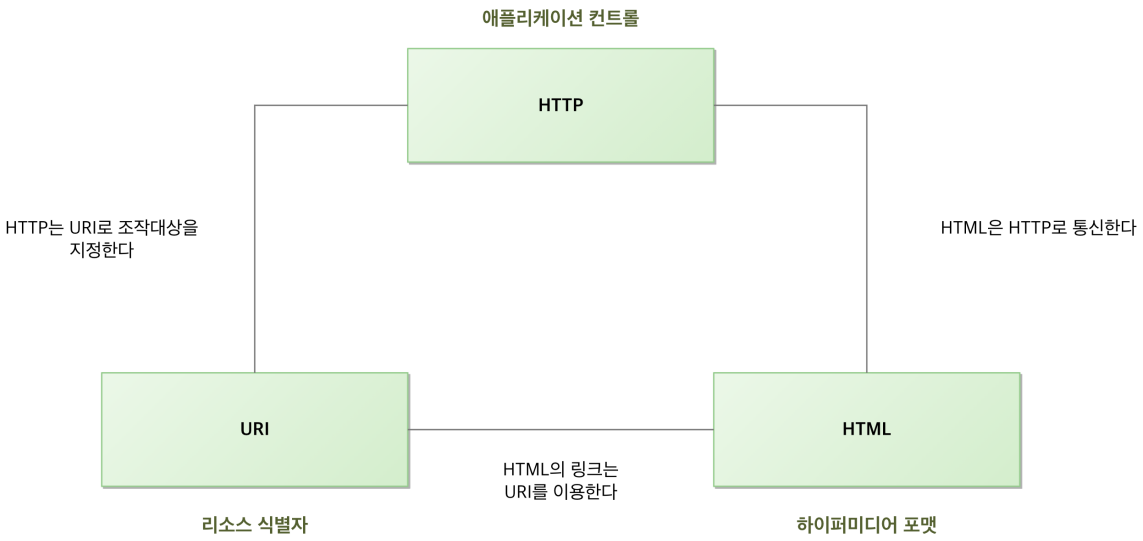
웹 브라우저에 대해서는 월드와이드웹 문서를 참고하십시오.

월드 와이드 웹(World Wide Web, WWW, W3)은 인터넷에 연결된 컴퓨터를 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간을 말한다. 간단히 **웹**(the Web)이라 부르는 경우가 많다. 이 용어는 인터넷과 동의어로 쓰이는 경우가 많으나 엄격히 말해 서로 다른 개념이다. 웹은 전자 메일과 같이 인터넷 상에서 동작하는 하나의 서비스일 뿐이다. 그러나 1993년 이래로 웹은 인터넷 구조의 절대적 위치를 차지하고 있다.

인터넷에서 HTTP 프로토콜, 하이퍼텍스트, HTML형식 등을 사용하여 그림과 문자를 교환하는 전송방식을 말하기도 한다.



WEB의 구성



URI

http:// blog.expamle.com/entries/1

- URI Scheme : http
- 호스트명 : blog.examples.com
- 패스 : /entries/1

http://harry:pass@kdt.programmers.com:8080/search?q=test&debug=true

- URI Scheme : http
- 사용자 : jeado:pass
- 호스트명 : blog.examples.com
- 포트번호 : 8080
- 패스 : /entries/1
- 쿼리 파라미터(쿼리 문자열) : q=test&debug=true

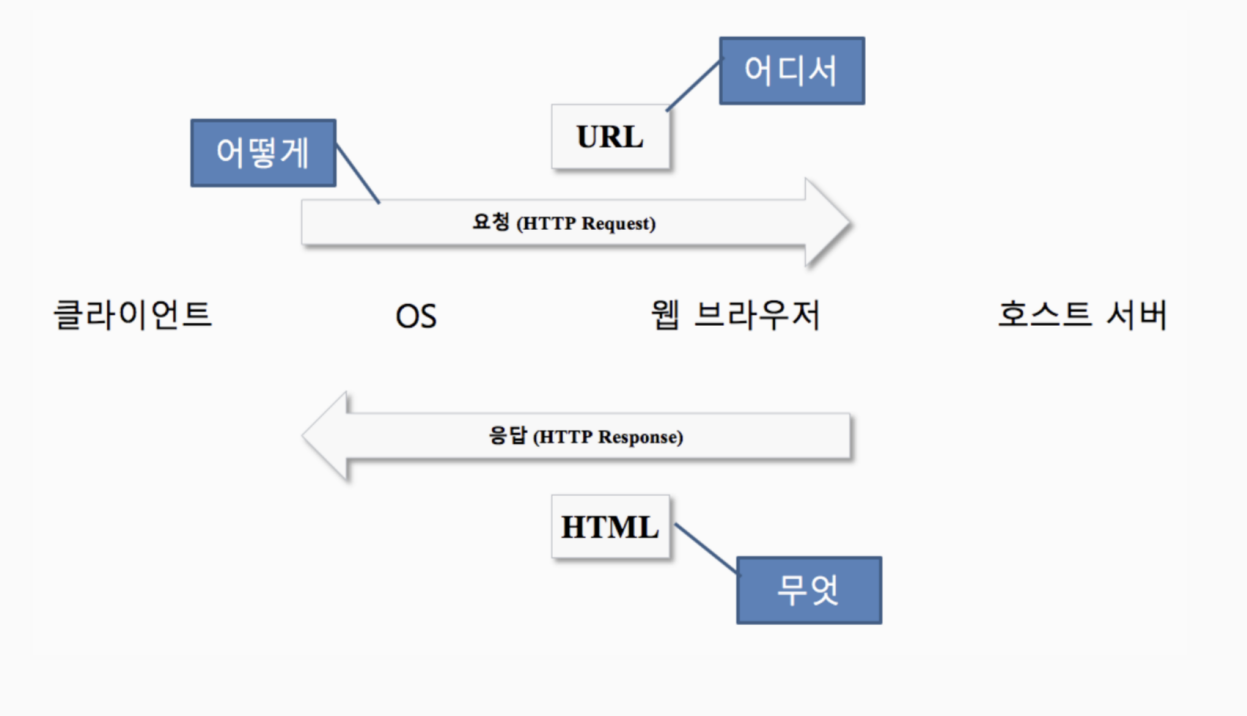
상대경로에서 절대 경로로의 변환 (시작점은 /foo/bar)

상대 경로	절대 경로
hoge	/foo/bar/hoge
hoge/fuga	/foo/bar/hoge/fuga
./hoge	/foo/bar/hoge
../hoge	/foo/hoge
../hoge/fuga	/foo/hoge/fuga
../../foge	/foge

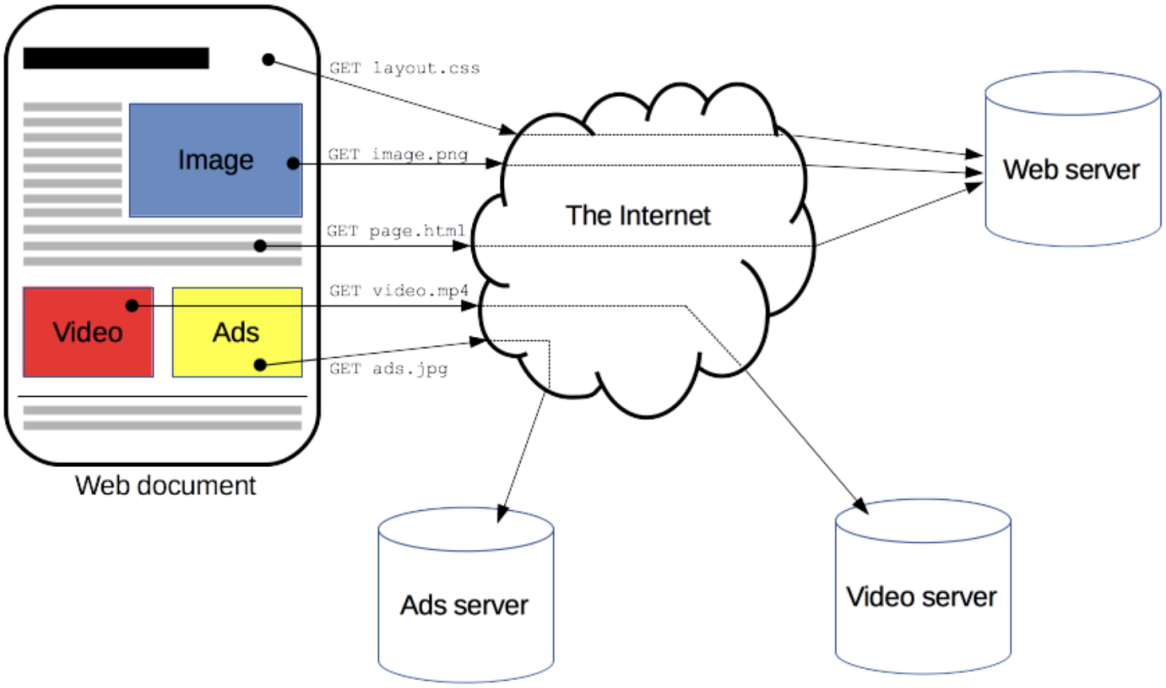
URI에서 사용할 수 있는 문자 (ASCII 문자)

- 알파벳 : A-Za-z
- 숫자 : 0-9
- 기호 : -.:~@!&'()

HTTP



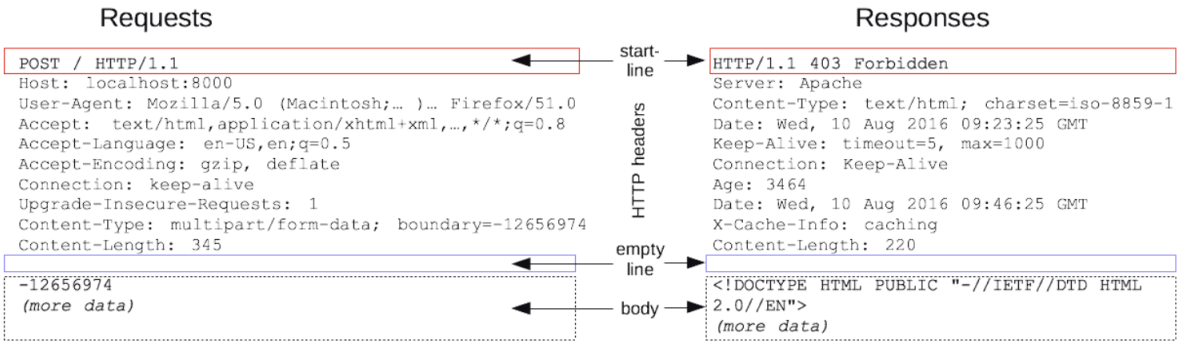
클라이언 - 서버



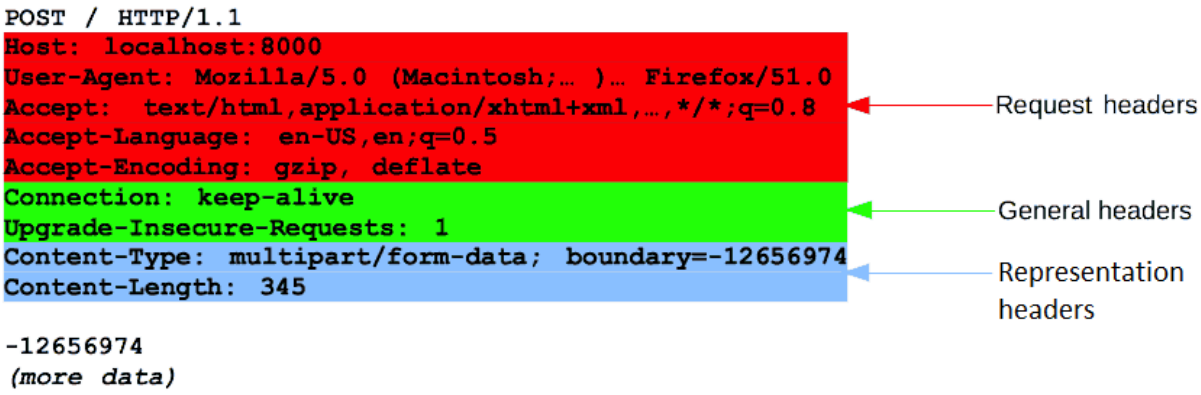
https://developer.mozilla.org/ko/docs/Web/HTTP/Overview

HTTP의 주요 특징

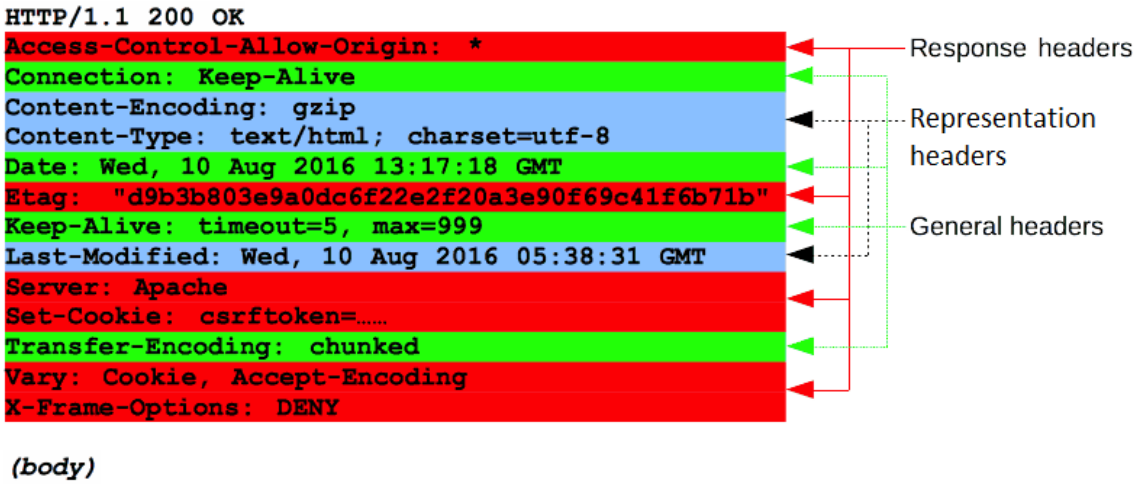
- TCP/IP 기반
- 요청/응답형 프로토콜
- 동기형 프로토콜
- 스테이트리스



https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages



출처: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages



출처: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages

HTTP Method

8개의 메서드

메서드	의미
GET	리소스 취득
POST	서브 리소스의 작성, 리소스 데이터 추가, 그 밖의 처리
PUT	리소스 갱신, 리소스 작성
DELETE	리소스 삭제
HEAD	리소스의 헤더 취득
OPTIONS	리소스가 서포트하는 메서드의 취득
TRACE	자기 앞으로 요청 메시지를 반환 시험
CONNECT	프록시 동작의 터널 접속으로 변경

HTTP 메서드와 CRUD

CRUD명	의미	메서드
Create	작성	POST/PUT
Read	읽기	GET
Update	갱신	PUT
Delete	삭제	DELETE

HTTP 개요 - HTTP | MDN

HTTP는 HTML 문서와 같은 리소스들을 가져올 수 있도록 해주는 프로토콜입니다. HTTP는 웹에서 이루어지는 모든 데이터 교환의 기초이며, 클라이언트-서버 프로토콜이기도 합니다. 클라이언트-서버 프로토콜이란 (보통 웹브라우저인) 수신자 측에 의해 요청이 초기화되는 프로토콜을 의미합니다. 하나의 완전한 문서는 텍스트, 레이

https://developer.mozilla.org/ko/docs/Web/HTTP/Overview

HTML

HTML (Hyper Text Markup Language) = Hyper Text + Markup

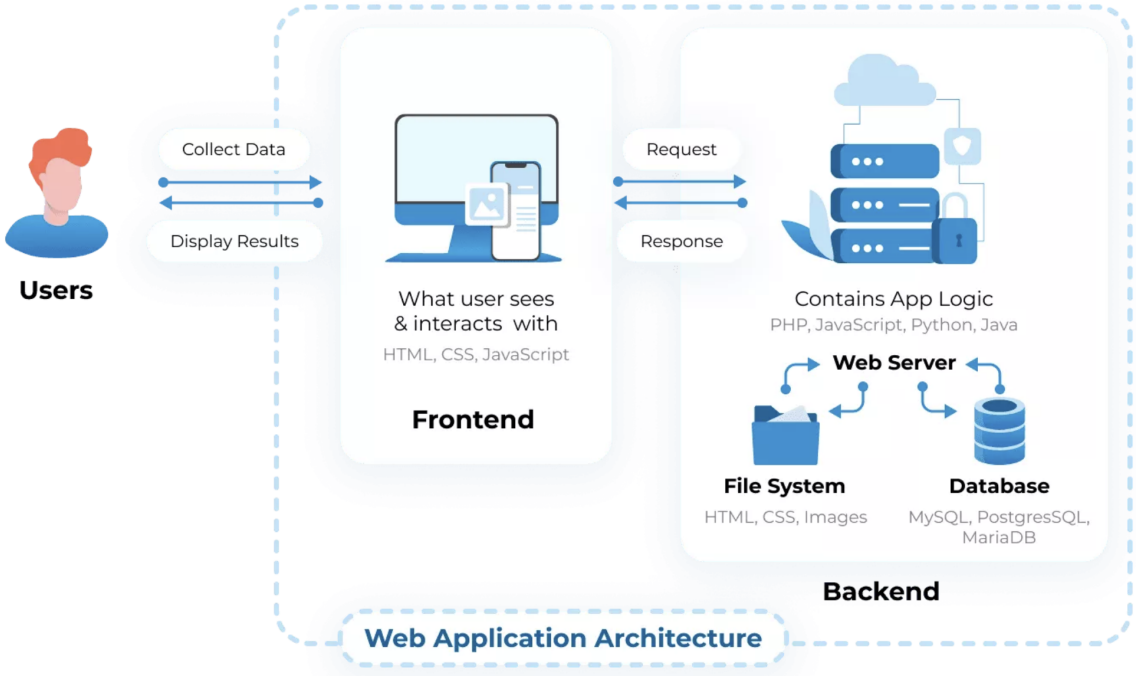
```
<!DOCTYPE html>
<html>
  <head>
    <title>How to put PHP in HTML - Simple Example</title>
  </head>
  <body>
    <h1>This message is from server side.</h1>
  </body>
</html>
```

웹의 기술적 특징 2가지

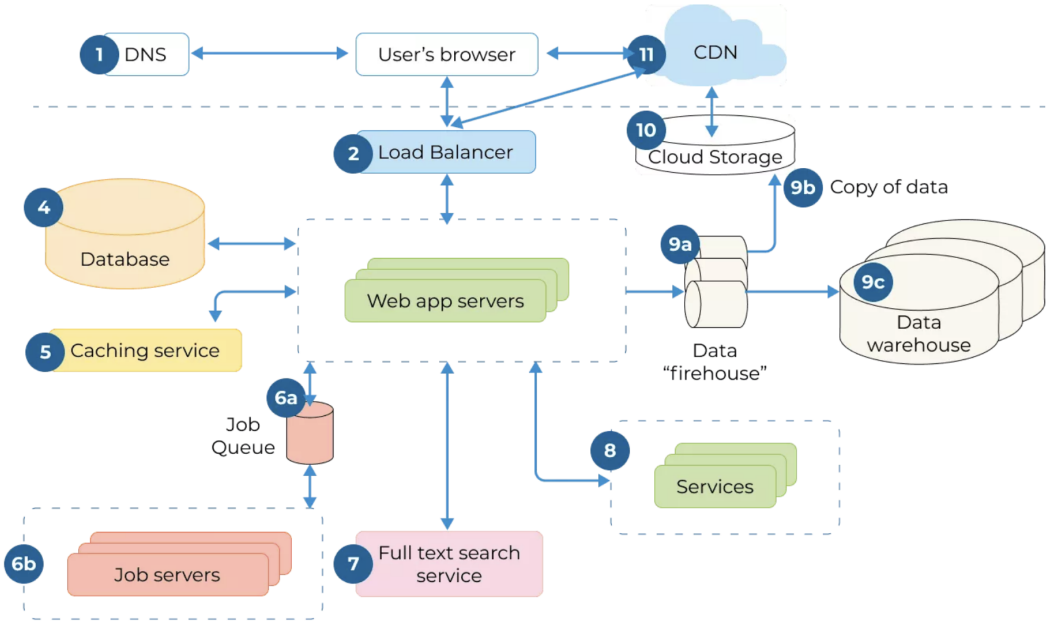
Hypermedia System (하이퍼미디어 시스템)

Distributed System (분산 시스템)

Web Application Architecture



https://litslink.com/blog/web-application-architecture

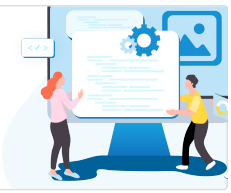


https://litslink.com/blog/web-application-architecture

Web Application Architecture: A Guide Through the Intricate Process of Building an App | LITSLINK Blog

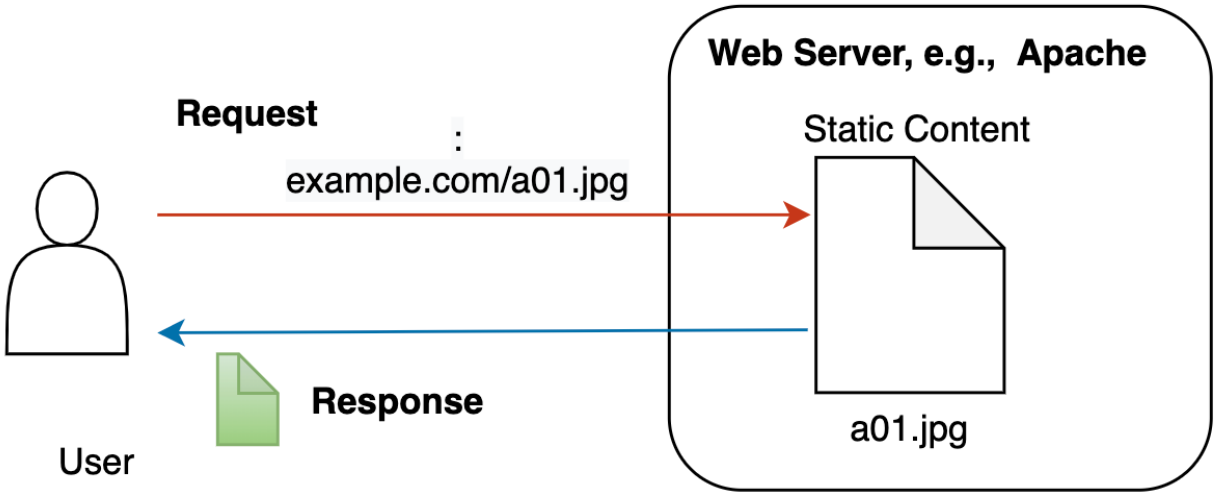
It's hard to imagine the world without the Internet today. More than 50 million people use it on a daily basis, and this figure is growing. Every entrepreneur willing to scale its business and make profit goes online. A website is a handy instrument that helps companies generate more traffic, attract customers and grow sales.

https://litslink.com/blog/web-application-architecture

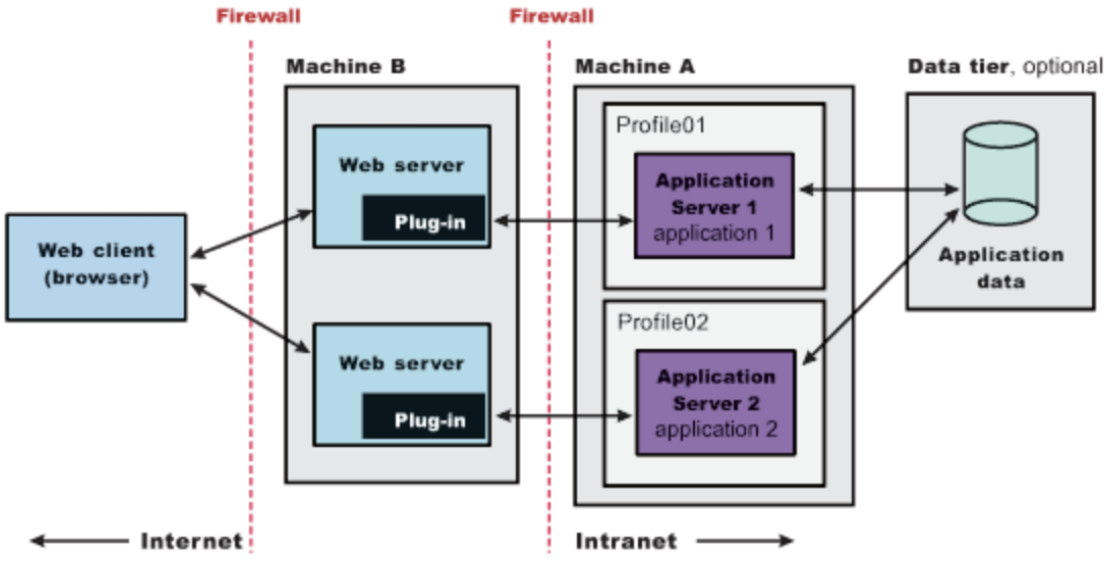


웹 서버 VS 웹 에플케이션 서버

Web Server

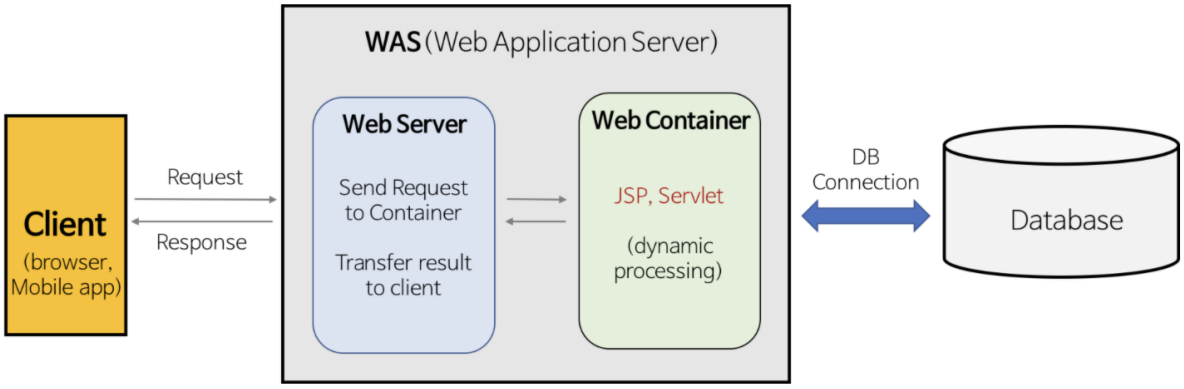


출처: https://mossgreen.github.io/Servlet-Containers-and-Spring-Framework/



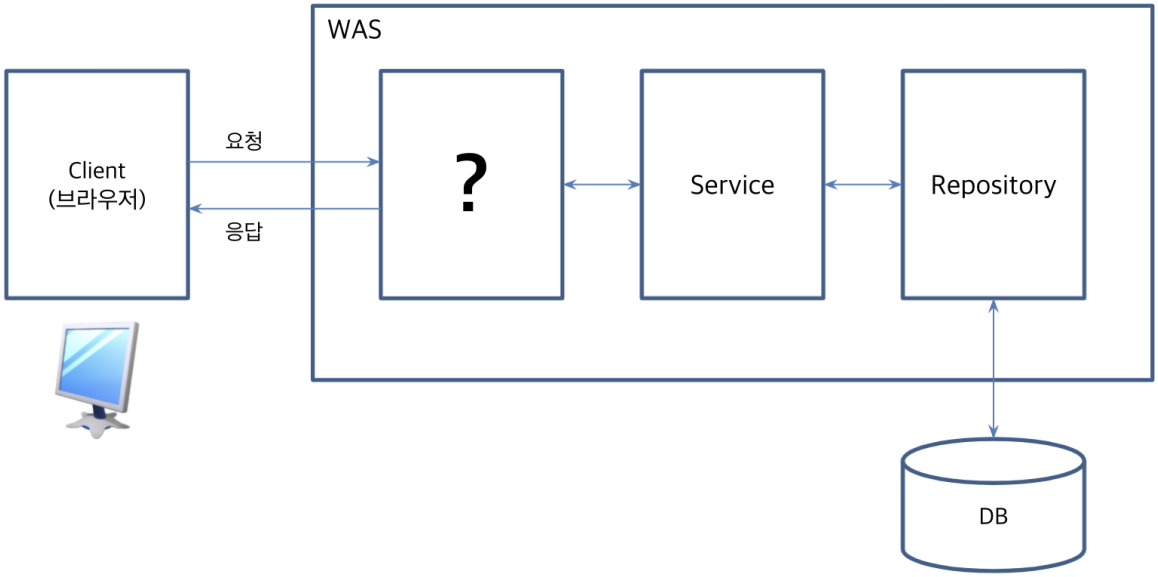
출처: http://setgetweb.com/p/WAS9/ae/tins_webplugins_mult_remotesa.html

Web Application Server



출처: https://gmlwjd9405.github.io/2018/10/27/webserver-vs-was.html

서블릿 이해하기



Jakarta Servlet

From Wikipedia, the free encyclopedia
(Redirected from Java servlet)

This article has multiple issues. Please help **improve it** or discuss these issues on the **talk page**. *(Learn how and when to remove these template messages)* [hide]

- This article **needs additional citations for verification**. *(February 2014)*
- This article **has an unclear citation style**. *(May 2016)*

A **Jakarta Servlet** (formerly Java Servlet) is a Java software component that extends the capabilities of a server. Although servlets can respond to many types of requests, they most commonly implement web containers for hosting web applications on web servers and thus qualify as a server-side servlet web API. Such web servlets are the Java counterpart to other dynamic web content technologies such as PHP and ASP.NET.

Jakarta Servlet

Original author(s)	Pavni Diwanji
Developer(s)	Eclipse Foundation
Initial release	December 1996; 24 years

https://en.wikipedia.org/wiki/Jakarta_Servlet

javax.servlet

Interface Servlet

All Known Subinterfaces:

`HttpJspPage`, `JspPage`

All Known Implementing Classes:

`FacesServlet`, `GenericServlet`, `HttpServlet`

`public interface` **`Servlet`**

Defines methods that all servlets must implement.

A servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.

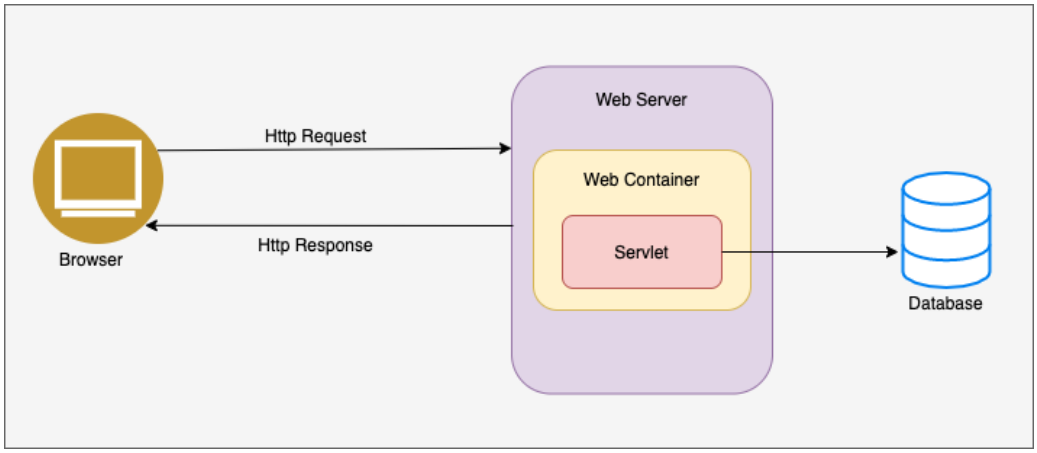
To implement this interface, you can write a generic servlet that extends `javax.servlet.GenericServlet` or an HTTP servlet that extends `javax.servlet.http.HttpServlet`.

This interface defines methods to initialize a servlet, to service requests, and to remove a servlet from the server. These are known as life-cycle methods and are called in the following sequence:

1. The servlet is constructed, then initialized with the `init` method.
2. Any calls from clients to the `service` method are handled.
3. The servlet is taken out of service, then destroyed with the `destroy` method, then garbage collected and finalized.

In addition to the life-cycle methods, this interface provides the `getServletConfig` method, which the servlet can use to get any startup information, and the `getServletInfo` method, which allows the servlet to return basic information about itself, such as author, version, and copyright.

https://javaee.github.io/javaee-spec/javadocs/javax/servlet/Servlet.html



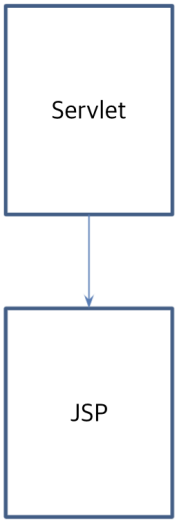
https://codeburst.io/understanding-java-servlet-architecture-b74f5ea64bf4


```
public class MyServlet extends HttpServlet {

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

    PrintWriter out = response.getWriter();
    out.println("<html>" +
                "<body>" +
                new java.util.Date() +
                "</body>" +
                "</html>");
  }
}
```

MyServlet.java

```
<servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>edu.test.MyServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>MyServlet</servlet-name>
    <url-pattern>/myservlet</url-pattern>
</servlet-mapping>
```

web.xml



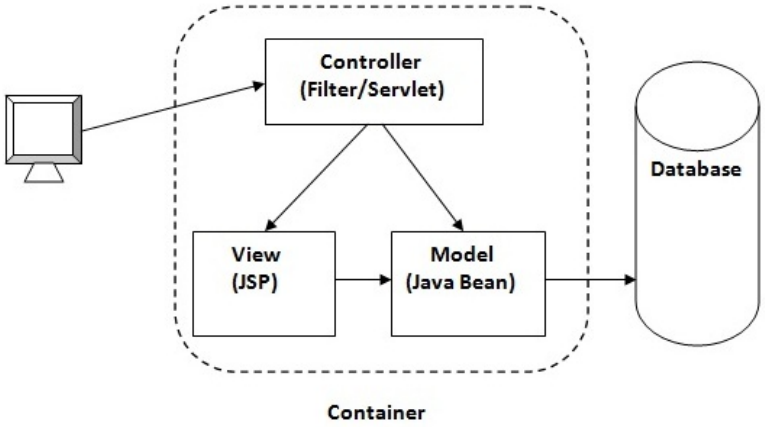

```
public class MyServlet extends HttpServlet {

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

    request.setAttribute("date", new java.util.Date());

    RequestDispatcher view = request.getRequestDispatcher("result.jsp");
    view.forward(request, response);
  }
}
```

MyServlet.java

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<body>
<%=request.getAttribute("date") %>
</body>
</html>
```

result.jsp

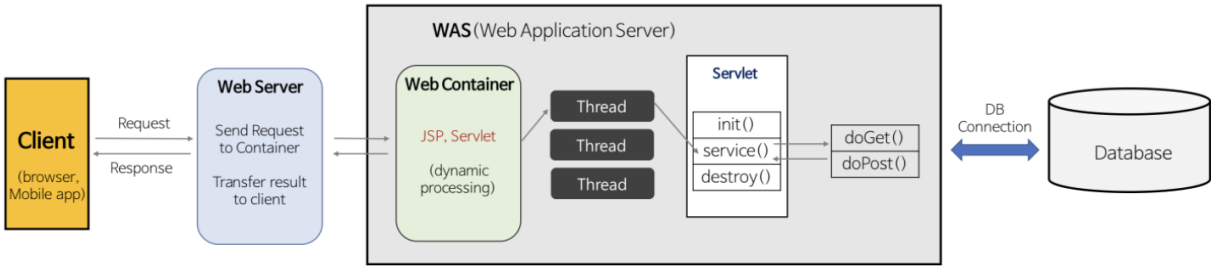
MVC Pattern in Servlet



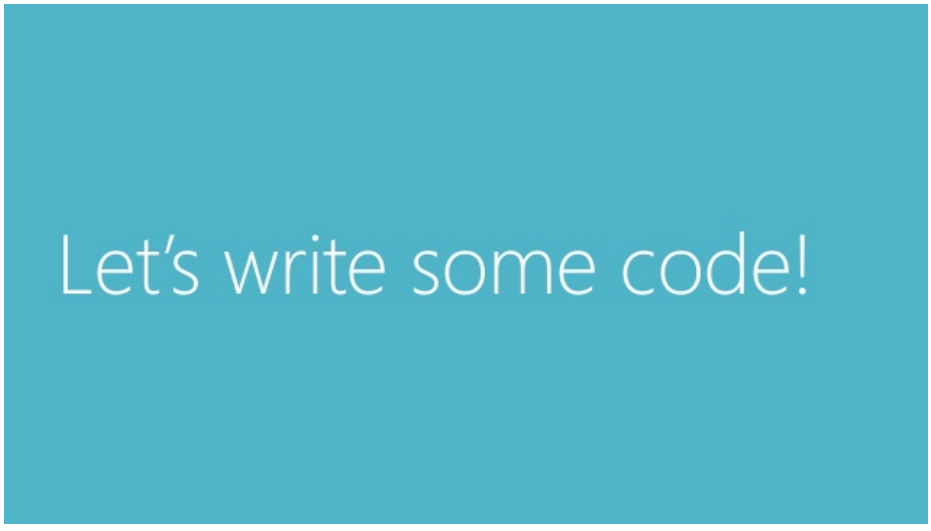
https://www.javatpoint.com/MVC-in-jsp

- Model - 모델은 자바 객체 또는 Pojo로 데이터를 담고 있습니다.
- View - 모델이 담고 있는 데이터를 시각적으로 보여주는 역할을 합니다.
- Controller - 모델과 뷰를 연결하기 위한 매개체입니다. 사용자의 입력/요청을 받아 모델의 상태를 변경해주고 그에 따른 뷰를 업데이트해 줍니다.

Servlet의 Life Cycle



https://gmlwjd9405.github.io/2018/10/27/webserver-vs-was.html



Head First Servlets and JSP, 2nd Edition

Servlets are controlled by the Container In chapter two we looked at the Container's overall role in a servlet's life-it creates the request and response objects, creates or allocates a ... - Selection from Head First Servlets and JSP, 2nd Edition [Book]

https://learning.oreilly.com/library/view/head-first-servlets/9780596516680/ch04s01.html