

서블릿

# Servlet / Jsp

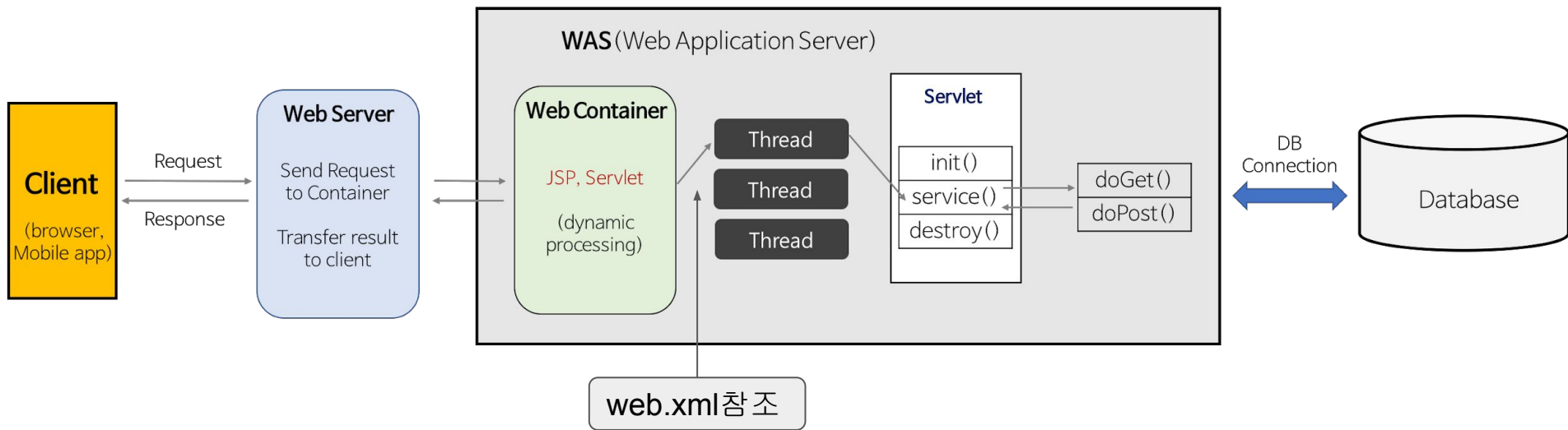
## Servlet

- JAVA 코드 내에 HTML 코드가 들어간다
- 웹페이지를 동적으로 수행하거나 데이터를 처리하기 위해 자바로 생성
- 코드 수정시 재 배포 해야 한다
- MVC패턴 기준 **Controller**으로 사용되어진다

## JSP

- HTML 코드 내에 JAVA 코드가 들어갈 수 있다.
- 코드 수정시 재배포 할 필요 없이 **WAS**가 처리 (JSP가 실행되면 WAS는 내부적으로 JSP 파일을 Java Servlet(.java)으로 변환한다.)
- MVC패턴 기준 **View**으로 사용 되어진다.

# Web Service Architecture



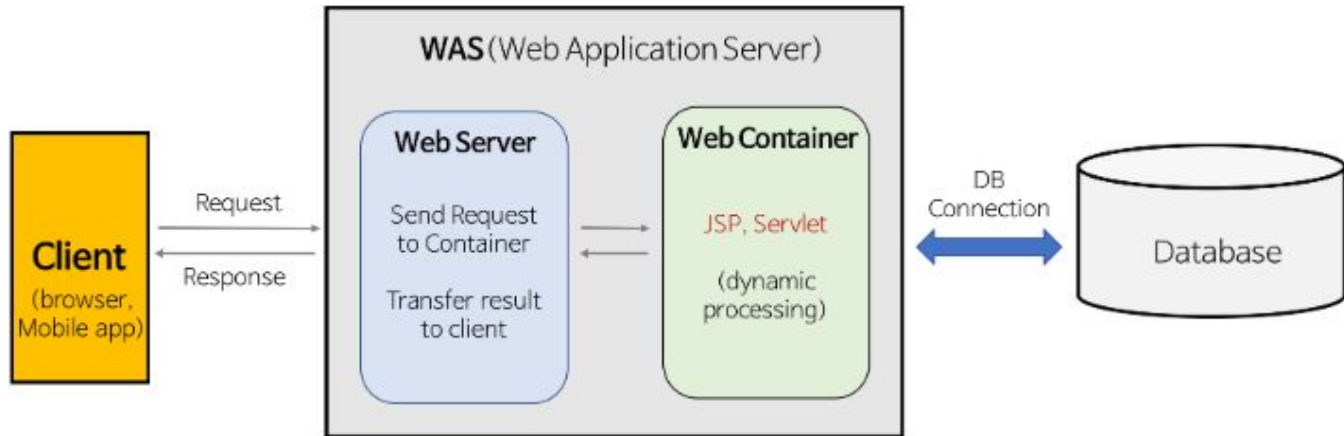
# 서블릿

- 서블릿 클래스로부터 만들어진 객체
- 웹 컨테이너는 서블릿 클래스를 가지고 서블릿 객체를 만든 후

그 객체를 초기화 시켜 웹 서비스를 할 수 있는 상태로 만든다. 이 과정을 거친 서블릿 객체만 서블릿이라 할 수있다.

**\*\*웹컨테이너란?**

웹 서버의 내부에서 서블릿 클래스 또는 **JSP** 파일을 실행하기 위한 실행 환경을 제공하는 역할



# 서블릿 클래스의 규칙

1. `public`클래스이어야 하고, `javax.servlet.http.HttpServlet`클래스를 상속하여야 한다

`HttpServlet <--상속-- GenericServlet <--구현-- Servlet`

2. `HttpServlet` 클래스 내부의 `doGet` 또는 `doPost` 메서드 내부에 요청이 왔을때 요청을 처리할 일을 기술해야 한다.(`override`)
3. HTML문서는 `doGet`, `doPost`메서드의 두번째 Parameter를 이용하여 출력하여야 한다.

ex) `doGet(HttpServletRequest req, HttpServletResponse resp)`

`HttpServletRequest`와 `HttpServletResponse`는 각각의 인터페이스를 `import` 해야한다.

4. 웹으로의 출력을 위한 `PrintWriter`클래스 호출

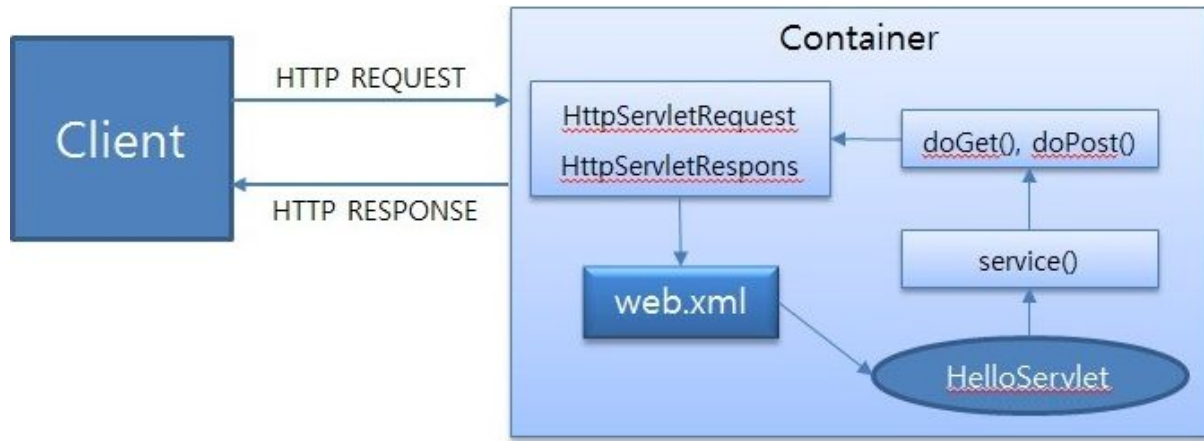
```
PrintWriter pw = resp.getWriter();
```

```
pw.println("<HTML>"); ←이런식으로 태그를 직접 넣을 수 있다.
```

# Servlet 생명주기

1. 서블릿이 처음 요청 된 경우 - 서블릿 클래스에서 메모리에 로딩하여 객체 생성
2. 객체가 생성되면서 `init()`메서드 호출하여 **Servlet** 초기화
3. **Thread**를 이용하여 **Service()**메서드가 호출 되어지고 `doGet`나 `doPost`메서드를 호출하여 처리한다
4. `destory()`메서드는 서블릿 컨테이너에 의해 호출되어 서블릿이 서비스를 중단하고 있음을 서블릿에 알립니다.
5. `init`과 `destroy`는 한번씩만

실행 되어진다.



# Servlet 생명주기

정보: Server startup in 230 ms

init()

service() 시작

doGet()

service() 종료

4월 24, 2019 6:59:35 오후 org.apache.catalina.core.StandardContext reload

정보: Reloading Context with name [/servletTest] has started

destroy()

정보: Server startup in 244 ms

init()

service() 시작

doGet()

service() 종료

service() 시작

doGet()

service() 종료

정보: Reloading Context with name [/servletTest] is completed

init()

service() 시작

doGet()

service() 종료

4월 24, 2019 7:00:40 오후 org.apache.catalina.core.StandardServer await

정보: A valid shutdown command was received via the shutdown port. Stopping the Server instance.

4월 24, 2019 7:00:40 오후 org.apache.coyote.AbstractProtocol pause

정보: Pausing ProtocolHandler ["http-nio-8080"]

4월 24, 2019 7:00:40 오후 org.apache.coyote.AbstractProtocol pause

정보: Pausing ProtocolHandler ["ajp-nio-8009"]

4월 24, 2019 7:00:40 오후 org.apache.catalina.core.StandardService stopInternal

정보: Stopping service [Catalina]

destroy()

# Servlet 컴파일

1. CMD창에서 컴파일 할 시 **import** 한 패키지가 **JDK** 표준라이브러리에 없다면 **-cp** 옵션을 이용하여 해당하는 라이브러리의 경로를 지정해주어야 한다.

ex)>javac -cp "D:\cafe24\apache-tomcat-8.5.40\lib\servlet-api.jar" servlet.java

2. 톰캣에서 필요한 jar파일을 jdk의 /jre/lib/ext쪽으로 복사한다

그 후

>javac servlet.java도 가능하다



# Servlet 설정

web.xml파일 내에

```
<servlet>
  <servlet-name>servTest</servlet-name>
  <servlet-class>sTest</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>servTest</servlet-name>
  <url-pattern>/test</url-pattern>
  <!-- <url-pattern>*.do</url-pattern> -->
</servlet-mapping>
```

servlet-mapping의 url-pattern에 맞는 경로가 들어오면 servlet-name를 찾아가서 그곳의 servlet-class를 찾아간다. 주석처럼 할 경우 \*.do 로 들어오는 모든 경로에 대응한다.

```
@WebServlet("/sTest")
public class sTest extends HttpServlet {
```

# HttpServletRequest, HttpServletResponse

## HttpServletRequest

- 요청을 보내는 쪽 정보를 담아온다.

ex) IP 정보, Cookie, Session, Method(GET, POST) 등

## HttpServletResponse

- 응답을 보내기 위한 정보를 담고있다.

HTTP 프로토콜을 따른다.

# HTTP프로토콜이란?


1. HTTP(Hypertext Transfer Protocol)는 인터넷상에서 데이터를 주고 받기 위한 **서버/클라이언트 모델**을 따르는 프로토콜이다.  
애플리케이션 레벨의 프로토콜로 **TCP/IP위에서 작동**한다. HTTP는 어떤 종류의 데이터든지 전송할 수 있도록 설계돼 있다.  
HTTP로 보낼 수 있는 데이터는 **HTML문서, 이미지, 동영상, 오디오, 텍스트 문서** 등 여러종류가 있다. 하이퍼텍스트 기반으로 (Hypertext) 데이터를 전송하겠다(Transfer) = 링크기반으로 데이터에 접속하겠다는 의미이다.
2. Connectless 방식으로 작동
  - Request후 Response 받으면 연결을 끊어버린다.

# URL 입력

## General

Request URL: <https://www.naver.com/>

Request Method: GET

Status Code:  200

Remote Address: 125.209.222.141:443

Referrer Policy: no-referrer-when-downgrade

# URL 입력

## Request Headers

**:authority:** www.naver.com

**:method:** GET

**:path:** /

**:scheme:** https

**accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3

**accept-encoding:** gzip, deflate, br

**accept-language:** ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7

**cache-control:** max-age=0

**cookie:** NNB=D7TXGEH5706FY; nx\_ssl=2; npic=8Pj/CWTfs3zk0INCo1EeuxXp8r5KLNuCCyrR71HUCVLw6I5g0b758dFMeF9uixylCA==; PM\_CK\_loc=a7a38faa6161387497c1b2cec9c1888123d4f

log.naver.com%2FPostView.nhn%3FblogId%3Dacornedu%26logNo%3D221128616501%26proxyReferer%3Dhttps%253A%252F%252Fwww.google.com%252F&r2=https%3A%2F%2Fwww.google.c

**upgrade-insecure-requests:** 1

**user-agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36

# URL 입력

## Response Headers

**cache-control:** no-cache, no-store, must-revalidate

**content-encoding:** gzip

**content-type:** text/html; charset=UTF-8

**date:** Wed, 24 Apr 2019 11:20:40 GMT

**p3p:** CP="CAO DSP CURa ADMa TAIa PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT DEM STA PRE"

**pragma:** no-cache

**referrer-policy:** unsafe-url

**server:** NWS

**status:** 200

**strict-transport-security:** max-age=63072000; includeSubdomains

**x-frame-options:** DENY

**x-xss-protection:** 1; mode=block