





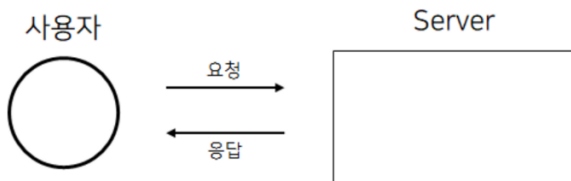


# 2023-하계방학 자기주도학습동아리 활동보고서

팀명	JUMP		
학습일시	2023년 07월 03일 (월요일)	진행차수	2주차
학습시간	14시 00분 ~ 16시 00분 (2시간)		
참석자	이 영 교 	김 태 현 	손 민 재 
	이 호 영 	최 재 신 	진 정 원 
불참자	X		
구 분	내 용		
학습목표	<p>서버와 클라이언트 간의 데이터를 공유하는 통신 방법에 대해 이해할 수 있다. Web Server 와 WAS 간의 차이점을 구분할 수 있다.</p>		
학습 방법 및 학습 내용	<p>1. 학습 방법 : 서버와 클라이언트 간의 통신 방법 중 Cookie와 Session에 대해 이해하고, 서버와 웹 서버의 차이점에 대해 이해한다. 추가로, 다른 조원들에게 알려주고 싶은 내용들을 정리하여 Notion Team Space 에 정리해서 올린다. 추가로, Github Organization 에 공부한 코드를 공유하며 일주일동안의 스터디 내용을 공유한다. 마지막으로, 디스코드에서 회의를 통해 2시간 동안 자신이 공부한 분량을 소개한다.</p> <p>** 1주차 회의록 노션 링크 : <a href="https://citrine-situation-94c.notion.site/2-2023-07-03-e4eba85b186b4d38b89c43cdc11333c9?pv=s=4">https://citrine-situation-94c.notion.site/2-2023-07-03-e4eba85b186b4d38b89c43cdc11333c9?pv=s=4</a></p> <p>** Github 링크 : <a href="https://github.com/JUMP-Hongik">https://github.com/JUMP-Hongik</a></p> <p>2. 학습 내용 : 1) Cookie와 Session을 사용하는 이유는 HTTP 프로토콜의 약점을 보완하기 위함으로, Connectionless 지향적인 현 통신 방법에서 이전의 정보들을 기억할 수 없기 때문에 로그인, 장바구니 등의 기능을 구현하는 과정에서 이전의 정보가 필요하다면 그 내용을 전달하는 기능을 하기 위함이다.</p> <div data-bbox="343 1556 917 1736">  <pre> graph LR     User((사용자)) -- 요청 --&gt; Server[Server]     Server -- 응답 --&gt; User </pre> </div> <p>그림1. HTTP 통신</p> <p>Connectionless 프로토콜은 클라이언트-서버가 한 번의 연결을 맺은 후, 요청에 대해 서버가 응답을 마치면 맺었던 연결을 끊어버리는 성질을 가지고 있다. 이렇게 통신을 하면 연결을 유지하기 위한 리소스를 줄일 수 있어 더 많은 사용자와 연결할 수 있다는 장점이 있다. 하지만 매번 새로운 연결을 시도/해제하는 과정에서 오버헤드가 발생한다.</p> <p>Stateless 프로토콜은 서버가 클라이언트의 상태를 보존하지 않는 것으로, HTTP 프로토콜의 방식이다. 항상 현재의 통신에만 집중하며 이전의 통신은 어떻게 했는지 상관하지 않는다. 따라서 이전의 내용을 기억해서 데이터를 제공해야 한다면, 클라이언트 측에서 추가로 데이터를 전송해야 한다는 단점이 있지만, 서버의 확장성이 높다는 측면에서 많이 사용하는 방법이다.</p>		



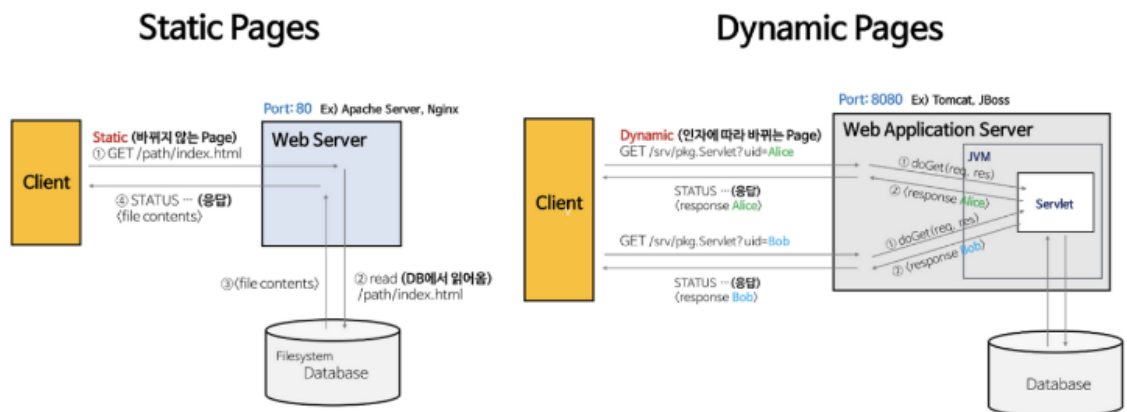
stateful 상태가 유지된다면, 항상 같은 서버가 유지되어야 한다는 큰 단점이 있다. 클라이언트 A의 요청을 서버 1이 기억하고 있다면, 항상 서버 1에게 요청을 보내야 한다는 것인데 서버 1에 장애가 발생한다면 통신을 할 수 없게 된다.

이 문제를 해결하기 위해 Stateless 상태의 통신 방법을 사용하게 된다면, 아무 서버에나 요청을 주고 받을 수 있으므로 필요한 데이터를 클라이언트에서 담아 전송하게 된다면 스케일 아웃, 즉 응답 서버를 쉽게 바꿀 수 있기에 그 확장성 또한 증가할 수 있다. 이 방법은 현재 통신 기법에 가장 적합한 방법이다.

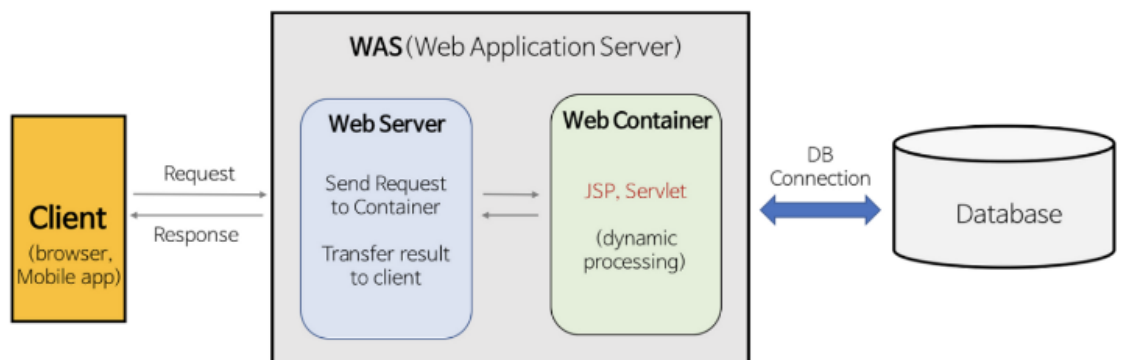
2) Cookie 는 웹 사이트에 접속할 때 생성되는 정보를 담은 임시 파일로, 서버가 클라이언트에 저장하는 데이터를 지칭한다. 상태 정보를 담아 클라이언트에서 저장하고, 이를 다시 필요할 때 서버에게 전달하는 과정을 거쳐 이전의 내용을 기억할 수 있게 만드는 것이다. 이름, 값, 만료일, 경로 정보 등으로 구성되며 300개의 쿠키를 저장할 수 있다. 도메인 당 20개의 쿠키까지 저장 가능하다는 특징이 있다.

### 3) Web Server vs WAS

웹 서버는 웹 브라우저에서 어떠한 페이지 요청을 하면 정적 콘텐츠를 제공하는 서버를 말한다. HTML, CSS, JS, 이미지 등 즉시 응답이 가능한 콘텐츠를 제공하며, 가장 앞단에서 클라이언트의 요청을 받아 전송하게 된다. 만약 동적 콘텐츠 요청이 온다면, WAS에게 요청을 넘기고 응답을 클라이언트에게 제공한다. Apache, Nginx 등이 Web Server 에 해당한다.

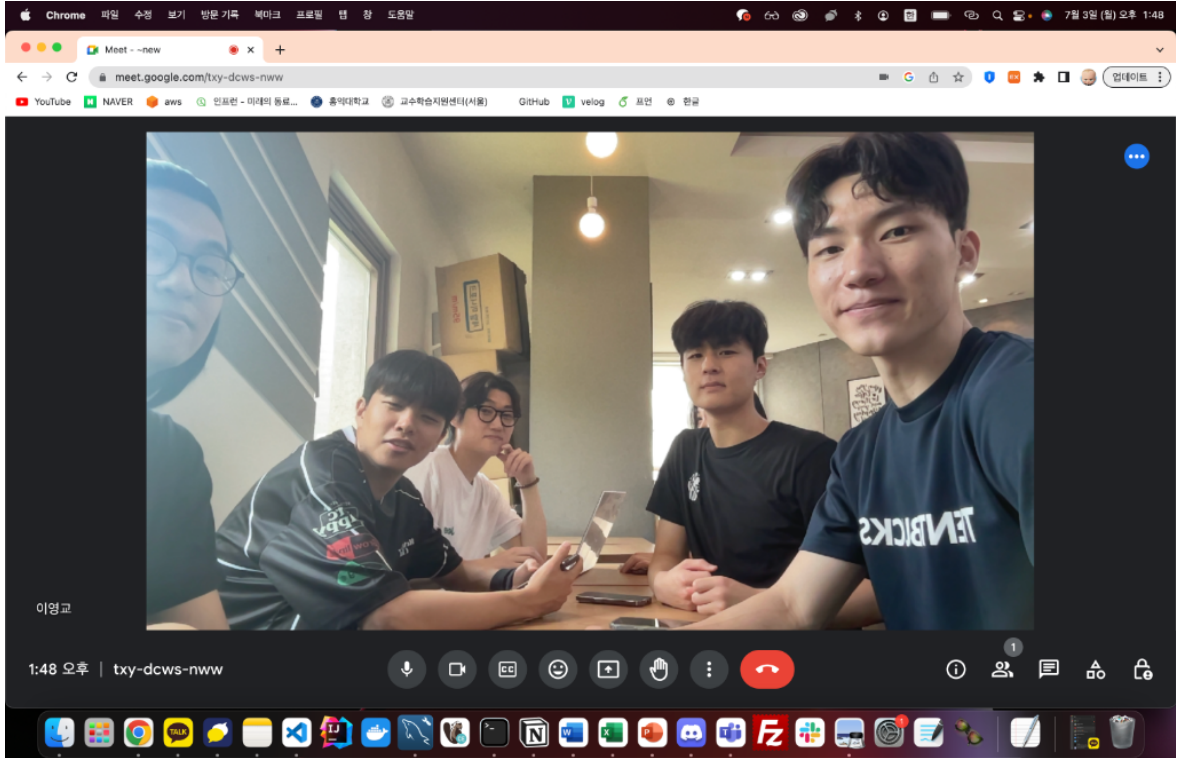


WAS 는 Web Server + Web Container 로 구성되며 웹 서버 단독으로 처리할 수 없는 DB 조회나 다양한 로직을 처리하는 동적 콘텐츠를 제공한다. JSP, Servlet 구동환경을 제공하며, 웹 상에서 사용하는 컴포넌트들을 올려놓고 사용하는 서버를 말한다. 우리가 실질적으로 서버라고 부르는 것은 이 WAS 를 말하는 것이다.

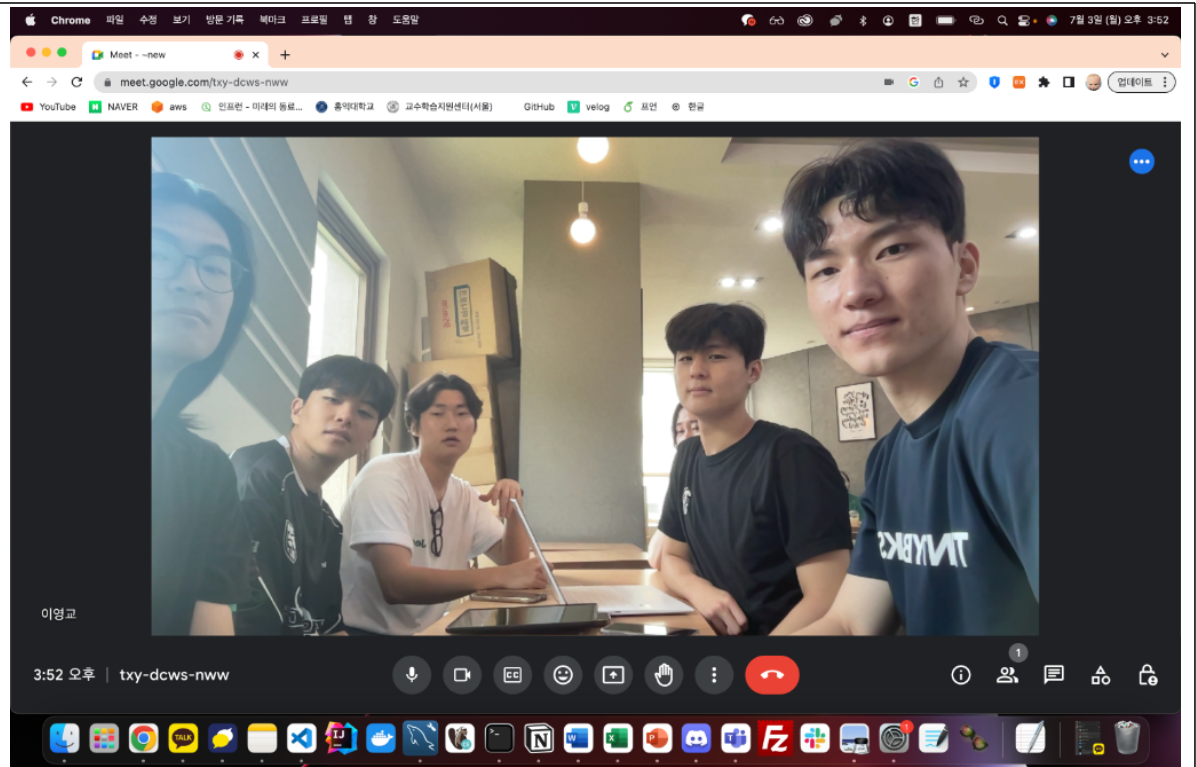


<b>활동 후기</b> * 본인이	팀장 / 팀원 이름	작성란
	이영교	팀장으로, 다른 팀원들의 회의 내용을 기록하며 해당 주차에 준비해온 프론트엔드 개발 지식들을 공유하는 역할을 했습니다



<p>맡은 역할 활동과 학습한 내용에 대한 후기 작성</p>		다. 스터디원의 내용을 수합하고, 보고서를 작성하는 등 내용을 한번 더 이해할 수 있어 좋습니다.
	김태현	2주차는 통신 기법과 JAVA 기본 내용에 대하여 복습하였습니다. 이전까지 했던 공부와는 다르게 학우들과 협업을 통해 이루고자 하는 목표가 있어 더욱 책임감 있게 공부할 수 있었습니다.
	손민재	Git, Github를 반복 숙달하면서, 프로젝트 협업 개발의 환경을 조금이나마 체험해보았습니다. 저학년이라 처음 공부해본 통신 기법을 이해하고 내용을 전달하는 것에 즐거움을 느꼈습니다.
	이호영	클라이언트 단과 서버 단을 구분하기 위해 /api를 추가하는 것에 대해서 이해하였고, 대면으로 공부하니까 더 집중이 잘 되는 부분이 있었다.
	최재신	통신 기법에 대해 다른 조원이 발표하는 내용을 듣고 추가로 현재 토큰이라는 방법이 있음을 공유하였다.
	진정원	새로운 내용들이 너무 많아 이해하기는 어려웠으나, 마지막까지 내용을 이해하려고 노력하였다.
<p>활동 증빙 사진</p>	학습시작	
		
	학습종료	





### 학습활동

