



“서버의 event 보내기

이진혁

”



“ 서버의 Event 보내기 ”

01

HTTP 맛보기

02

Polling

03

Long Polling

04

Web Socket

05

SSE

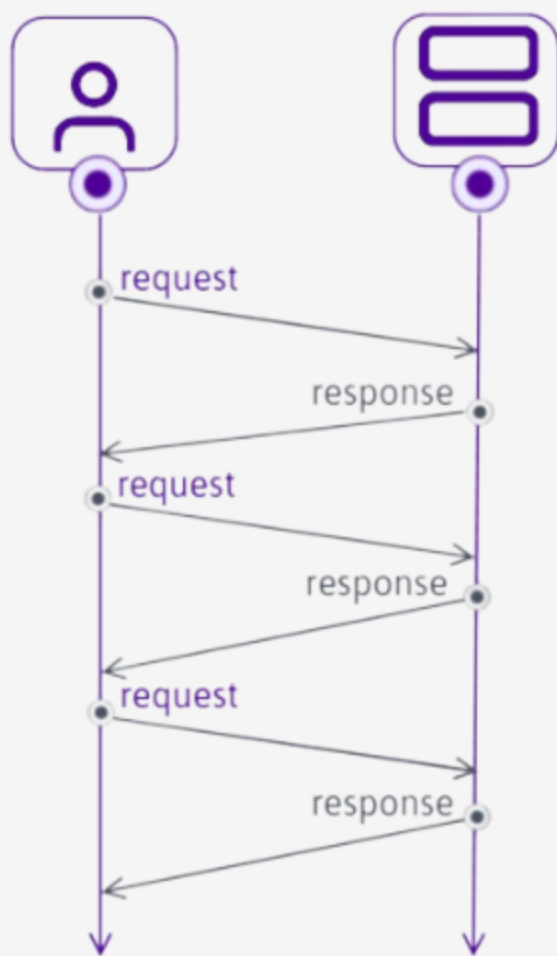


“

HTTP

”

단방향, Stateless



1. 클라이언트-서버 모델

- HTTP는 클라이언트와 서버간의 요청-응답 모델을 기반으로 동작합니다. 클라이언트가 요청을 보내고 서버가 응답을 반환

2. 무상태 프로토콜(Stateless Protocol)

- HTTP는 상태를 유지하지 않습니다. 각 요청은 독립적이며, 이전 요청과의 연결 정보를 갖지 않음

3. 텍스트 기반 프로토콜

- HTTP 메시지는 인간이 읽을 수 있는 텍스트 형식으로 구성되어 있습니다. HTTP 요청과 응답 메시지는 헤더와 본문으로 나뉨

4. TCP 기반 프로토콜

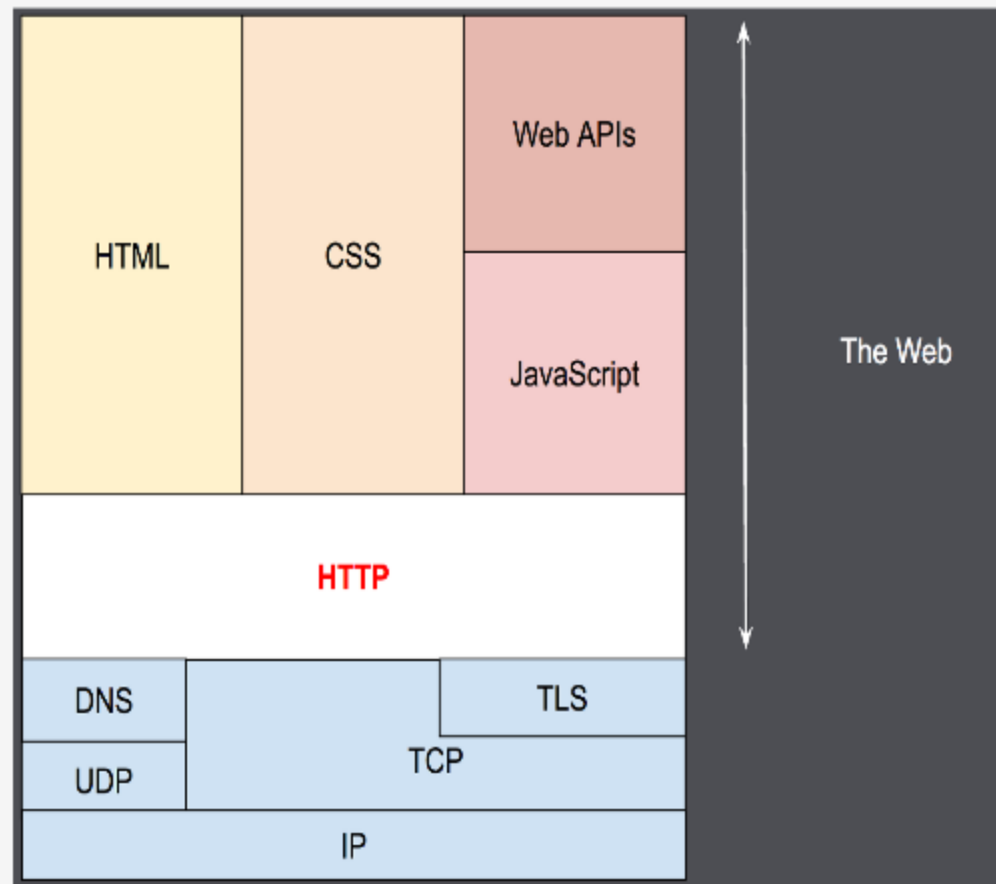
- connect, disconnect은 TCP 레벨에서 이루어지고 transmit은 HTTP 레벨에서 이루어짐

“

조금만 더 깊게

”

단방향?? 양방향??



HTTP는 TCP 기반 프로토콜

TCP는 기본적으로 포트 번호를 할당받아 TCP 레이어가 데이터의 목적지인 애플리케이션을 식별하는 Socket Connection

그렇다면 HTTP는 양방향 통신인가??

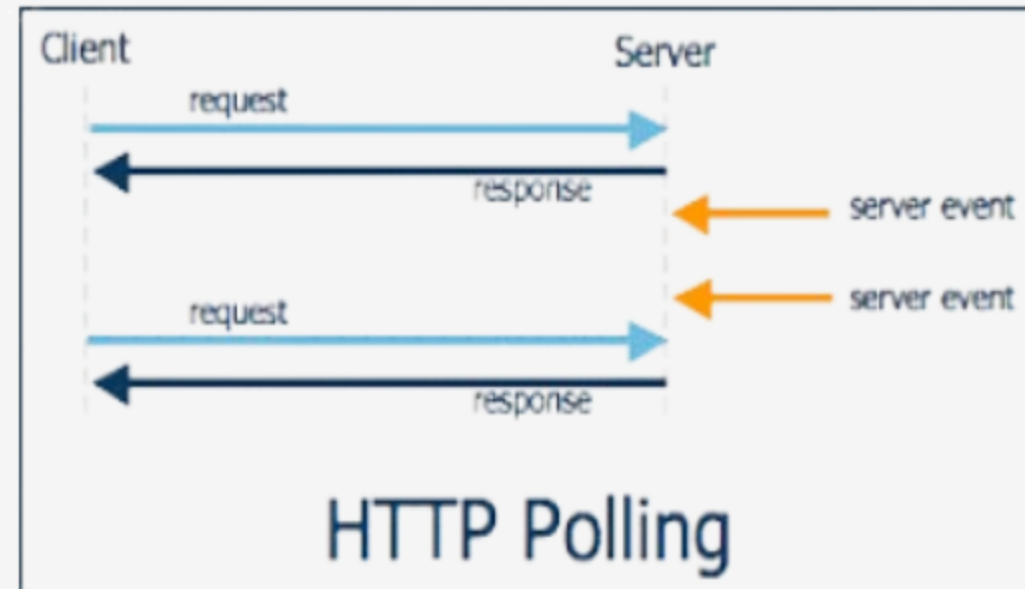
- 요청이 있어야 응답이 있기에(주체가 명확하기에) 그리고 동작방식이 양방향으로 이루어지지 않기 때문에 우리는 보통 단방향 통신이라고 함

“

Polling

”

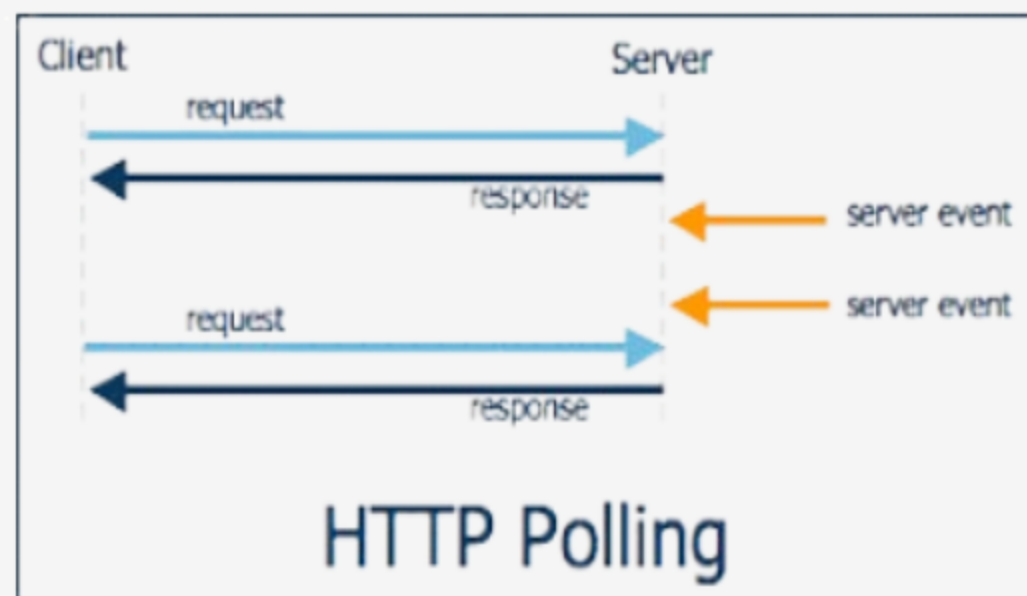
주기적 요청



Server에서 어떠한 event가 언제 발생할지 예측이 불가능하기 때문에 클라이언트가 평범한 HTTP 요청을 일정한 주기로 서버에 요청하여 이벤트 내용을 전달받는 방식

“ Long Polling ”

주기적 요청

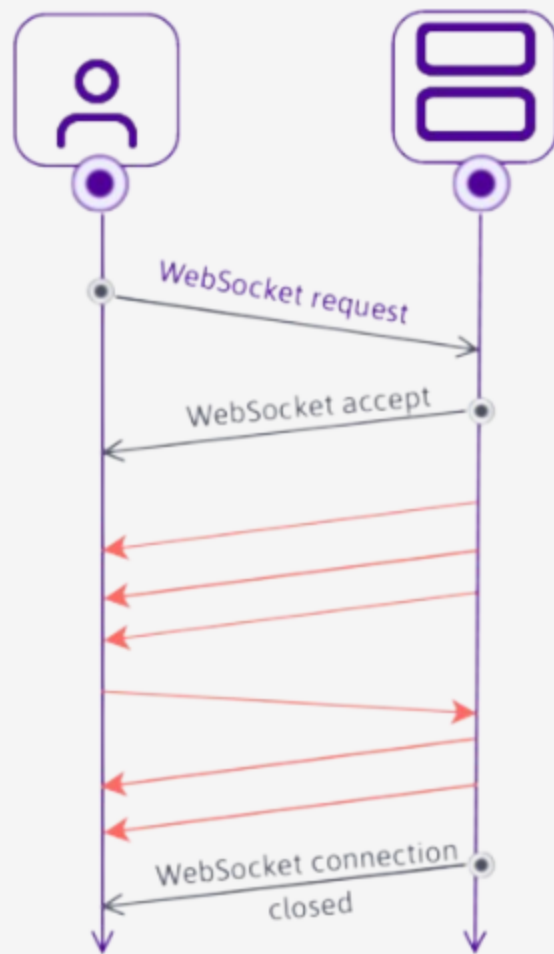


클라이언트에서 서버로 일단 http 요청을 보내고 이 상태로 계속 기다리다가 서버에서 해당 클라이언트로 전달할 이벤트가 있다면 그 순간 response를 전달하며 연결이 종료된다. 해당 작업이 완료된 이후에는 클라이언트에서 곧바로 다시 http 요청을 보내 서버의 다음 이벤트를 기다리게 되는 작업 방식

- 많은 양의 메시지가 쏟아지면 polling과 많이 차이 나지 않음

“ Web Socket ”

주기적 요청



클라이언트와 서버를 연결하고 실시간으로 통신이 가능하도록 하는 첨단 통신 프로토콜이다. 웹소켓은 하나의 TCP 접속에 전이중(duplex) 통신 채널을 제공한다. 쉽게 말해, 웹소켓은 Socket Connection을 유지한 채로 실시간으로 양방향 통신 혹은 데이터 전송이 가능한 프로토콜

- Socket과의 차이점
 - 일반 HTTP Request 를 통해 Handshaking 과정을 거쳐 최초 접속이 이루어짐
 - 어플리케이션 레이어에서 Handshaking 이 일어난다는 의미

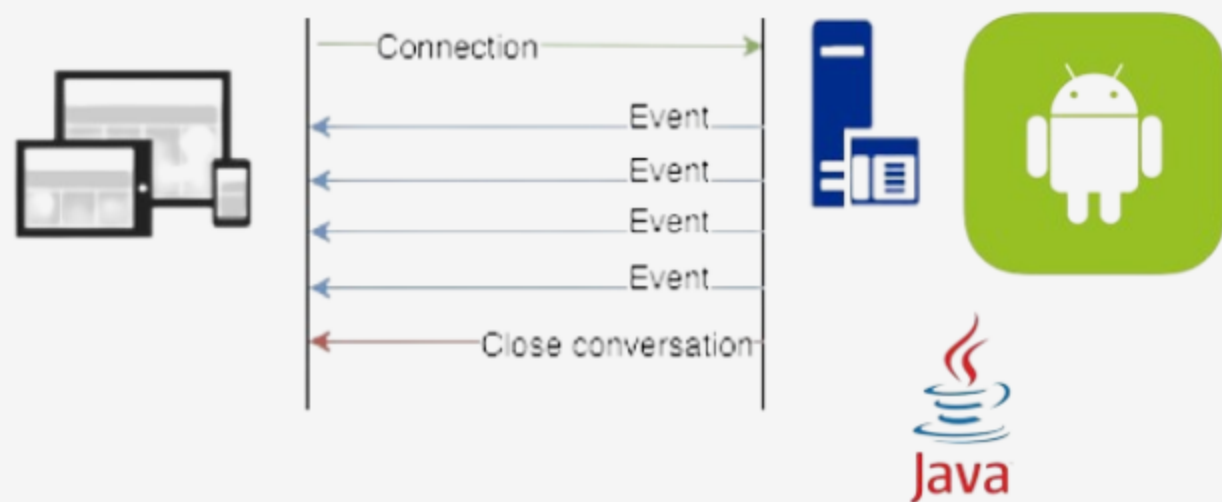
“

SSE

”

주기적 요청

Server-Sent Event (HTTP)



3 Way-HandShake로 connection을 열고 서버에서 event가 발생할 때
마다 text/event-stream 타입으로 서버에서 클라이언트로 단방향 통신을 할
수 있는 프로토콜

“

적용하기

”

목표 몸무게 수정 및 자신의 몸무게를
일주일 동안 수정하지 않은 사람에게 알
림을 보내려고 한다. 이 중 어떤 것을 사
용하면 좋을까?

“

적용하기

”

SSE

