

실시간 통신 방법

Web Socket

- 양방향 통신을 위해 도입된 프로토콜
- http 프로토콜이 아닌 ws 프로토콜 사용(즉, 새로운 프로토콜)

그렇다면 http 프로토콜을 이용한 실시간 소통 방법은?

Polling

- 클라이언트에서 주기적으로 요청을 보내는 방법
 - 그러나 실제로 실시간이 아님(주기적으로 요청과 응답을 주고 받는 시간 단위를 줄이면 사용자 입장에서는 실시간처럼 보이기 때문)
 - 그리고 주기적으로 요청과 응답을 보내야 하여 서버에 부하가 생김

Long Polling

- Polling 을 보완
- 클라이언트에서 요청을 보내고 서버에서 응답이 올 때까지 기다림
- 서버에서 변화가 생겨서 응답을 보내면
- 그 즉시 클라이언트에서 다시 요청을 보내고 대기하는 방법
 - 그러나 요청과 응답이 여전히 1대1 이라는 점에서 부하가 발생

SSE(Server Sent Events)

- 단 한번의 요청으로 연결을 지속하고 서버측에서 변화가 생길 때마다

- 응답을 반복해서 보낼 수 있음
- 완전히 실시간 ⇒ 서버에서 이벤트가 발생할 때마다 실시간으로 메시지를 보낼 수 있기 때문에 실시간성이 훨씬 뛰어남
- Polling 기법 보다 훨씬 적은 통신 횟수를 가지기 때문에 비용 절감
- 새로운 프로토콜을 익힐 필요가 없음(새로운 기술 도입 시 러닝 커브가 중요한 변수이기 때문)
- 양방향으로 소통을 하는 웹소켓 보다 비용쪽에서 우월

이외의 방법 Firebase

FCM

- 무료로 메시지를 전송할 수 있는 교차 플랫폼 메시징 솔루션
- 플랫폼에 종속되지 않고 푸시 메시지를 전송
- iOS, Android, Web 각 플랫폼에서 푸시 메시지를 보내려면 각 플랫폼 환경별로 개발해야하는 불편함이 있는데 이를 하나로 통합

Real Time Database

- 실시간 데이터베이스는 실시간으로 데이터를 저장 및 동기화해주는 NoSQL 데이터베이스