

# Web Socket

≡ 태그

[WebSocket?](#)

[탄생 배경](#)

[웹 소켓과 HTTP의 다른 점](#)

[예제 구현](#)

[출처](#)

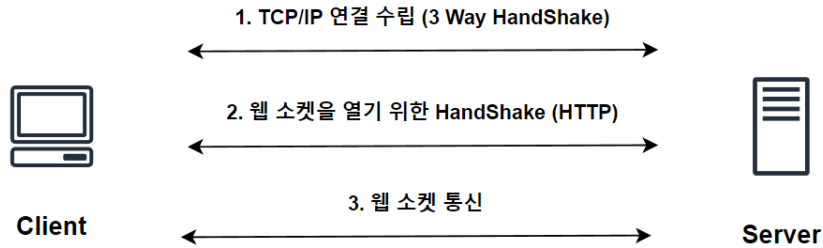
## WebSocket?

WebSocket(웹 소켓)이란 HTTP 환경을 기반으로 하여 TCP/IP 연결을 통해 전이중 통신(양방향 송수신) 채널을 제공하는 컴퓨터 통신 프로토콜이다. WebSocket의 접속 과정은 TCP/IP 접속, 웹 소켓을 열기 위한 HandShake 과정으로 나뉘 수 있다.

- 실시간 알림, 실시간 채팅 등 실시간이라는 키워드가 들어가는 기능에는 대부분 웹 소켓 기술이 쓰인다.
- HTML5 표준 기술로 HTTP 환경에서 클라이언트와 서버 사이에 하나의 TCP 연결을 통해 실시간으로 전이중 통신을 가능하게 하는 컴퓨터 통신 프로토콜이다.
  - 전이중 통신이란, 송신이나 수신만 가능한 단방향 통신이 아닌 양방향으로 실시간 송신, 수신이 가능한 통신을 말한다. Ex) 전화
- 이전에는 Polling이나 Long Polling으로 실시간 통신에 준하게 기능을 구현했지만 지금은 WebSocket을 사용한다.

*폴링 또는 인터로깅은 클라이언트 프로그램에서 동기식 활동으로 외부 장치의 상태를 능동적으로 샘플링하는 것을 말합니다. 폴링은 입출력(I/O) 측면에서 가장 자주 사용되며, 폴링된 I/O 또는 소프트웨어 구동 I/O라고도 합니다. 하드웨어 구현의 좋은 예로는 감시 타이머 (watchdog)가 있습니다. 출처*

WebSocket 최초 연결 과정을 나타내면 아래와 같다.



1. 서버와 클라이언트 간 TCP/IP 연결을 수립한다.
2. HTTP 요청 기반의 HandShake 과정을 거친다. (HTTP Request 헤더에 Upgrade, Connection를 추가하여 WebSocket 요청임을 표시)
3. HandShake 연결이 끝나면, HTTP 프로토콜을 WebSocket 프로토콜로 변환하여 웹 소켓 통신을 시작한다.

## 탄생 배경

초기 웹의 탄생 목적 : 문서 전달, 하이퍼링크를 통한 문서 연결.

- HTTP 프로토콜은 이러한 초기 웹의 탄생 목적에 부합하는 프로토콜이다.

하지만 웹에 동적인 표현과 뛰어난 상호작용이 요구되었음. 여러 새로운 기술이 만들어졌는데 그 중 하나가 웹 소켓이다. 실시간 양방향 통신을 위해 만들어졌다.

- 이전에는 Polling이나 Long Polling으로 실시간 통신에 준하게 기능을 구현하였음.

웹 소켓은 HTTP 프로토콜을 기반으로 Handshake 과정을 거쳐 연결을 수립한다.

웹 소켓은 연결을 수립하기 위해 Upgrade 헤더와 Connection 헤더를 포함하는 HTTP 요청을 보낸다.

```

GET /spring-websocket-portfolio/portfolio HTTP/1.1
Host: localhost:8080
Upgrade: websocket ①
Connection: Upgrade ②
Sec-WebSocket-Key: Uc9l9TMkWGbHFD2qnFHltg==
Sec-WebSocket-Protocol: v10.stomp, v11.stomp
Sec-WebSocket-Version: 13
Origin: http://localhost:8080
  
```

YAML

```

HTTP/1.1 101 Switching Protocols ①
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: 1qVdfYHU9hP0l4JYYNXF623Gzn0=
Sec-WebSocket-Protocol: v10.stomp
  
```

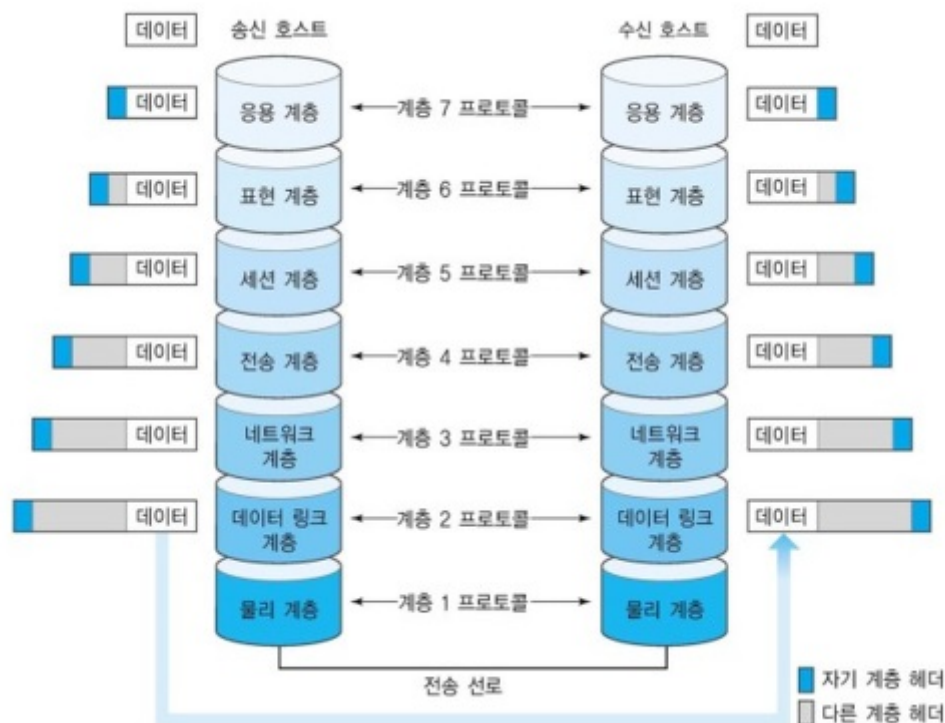
YAML

웹 소켓 연결 수립 과정에서의 응답 메시지는 일반적인 HTTP 응답 메시지와 다르다.

- 101 Switching Protocols: Handshake 요청 내용을 기반으로 다음부터 WebSocket으로 통신할 수 있다.
- Sec-WebSocket-Accept: 보안을 위한 응답 키

Handshake 과정을 통해 연결이 수립되면 응용 프로그램 계층 프로토콜이 HTTP에서 웹 소켓으로 업그레이드된다.

- 네트워크 계층 (OSI 7계층) : 물리 / 데이터 링크 / 네트워크 / 전송 / 세션 / 표현 / 응용



## 웹 소켓과 HTTP의 다른 점

웹 소켓은 HTTP 프로토콜 위에서 작동하지만 HTTP 프로토콜과 다르다.

- HTTP는 클라이언트와 서버간 접속을 유지하지 않는다. 요청과 응답 형태로 단방향 통신만 가능하다.
  - 클라이언트 → 서버 방향으로 요청을 보내는 건 가능하지만 서버 → 클라이언트 방향으로 요청을 보내는 건 불가능하다.
  - 요청 - 응답이 완료되면 HTTP 연결은 종료된다.

- WebSocket은 요청-응답 개념이 없으며, 서로 데이터를 주고 받으며 연결을 유지한다.
- REST 방식의 HTTP 통신은 URI와 HTTP Method를 통해서 웹 어플리케이션과 상호 작용하지만 웹 소켓은 초기 연결 수립을 위한 하나의 URL만 있으면 된다. 연결 수립 이후에는 TCP 연결로만 통신한다.

## 예제 구현

추후에 해보기

## 출처

### Polling (computer science)

Polling, or interrogation, refers to actively sampling the status of an external device by a client program as a synchronous activity. Polling is most often used in terms of input/output (I/O), and is also referred to as polled I/O or software-driven I/O. A good example of hardware implementation is a watchdog timer.

₩ [https://en.wikipedia.org/wiki/Polling\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Polling_(computer_science))

### 웹 소켓에 대해 알아보자! - 테스트 편



이 글은 웹 소켓으로 만든 채팅방을 어떻게 테스트할지 고민하는 독자를 위해 작성되었다. 웹 소켓은 어떻게 테스트하면 좋을까?

Babble...

<https://tecoble.techcourse.co.kr/post/2021-11-10-web-socket-test/>

### [WebRTC] ReactJS + EC2 + OpenVidu 환경에서 화상회의 구현

WebRTC (Web Real-Time Communication)란 별도의 플러그인이나 소프트웨어 없이 실시간으로 데이터(음성, 영상, 텍스트, 파일)를 서버(중계자) 없이 브라우저 간에 교환할 수 있도록 하는 기술이다. WebRTC 기술은 다

₩ <https://velog.io/@ohjinseo/ReactJS-EC2-OpenVidu-환경에서-화상회의-구현해보기>

