

실시간
채팅 구현
20304 신유민

목차

01

WebSocket과
Polling의 차이점

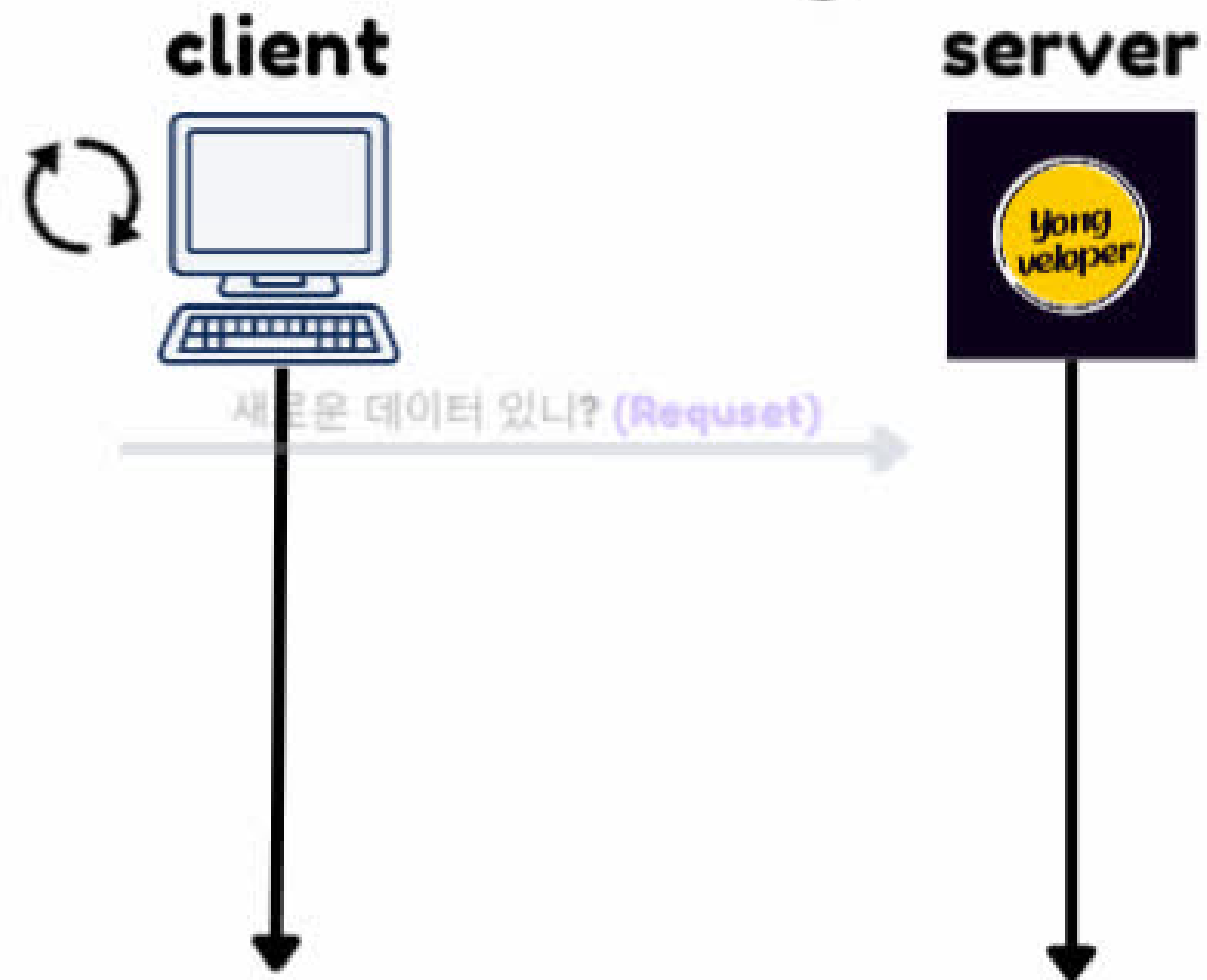
02

코드설명

03

소감 및 Q&A

Polling



Web Socket

client



server



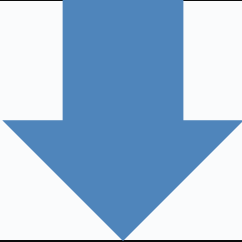
우리 소켓 연결하자 (HTTP Upgrade)

ㅇㅋ (HTTP 101 Response)

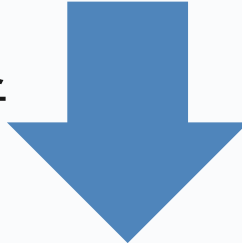


코드 설명

```
PS C:\Users\LG> mkdir ch-app
```

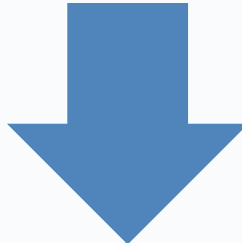


```
PS C:\Users\LG> cd ch-app  
PS C:\Users\LG\ch-app> npm init -y
```



npm = Node.js를 더욱 편리하게 사용할 수 있도록 만드는 핵심 요소

```
PS C:\Users\LG\ch-app> npm install express socket.io
```



```
PS C:\Users\LG\ch-app> code .
```

코드 설명

```
{ } package-lock.json
```

```
{ } package.json
```

코드 설명

```
{ } package-lock.json
```

```
{ } package.json
```



JSON

자바 스크립트 객체 표기법

전체코드

```
<html lang="ko">
<body>
<div class="chat-container">
  <div class="user-info">
    <strong>닉네임:</strong> <input type="text" id="nameBox" value="익명" style="width: 80px;">
  </div>
  <div class="chat-box" id="chatBox"></div>

  <div class="input-area">
    <textarea id="textBox" placeholder="메시지를 입력하세요..."></textarea>
    <button onclick="send()">전송</button>
  </div>
</div>

<script>
const socket = io();

// 1. 메시지 전송 함수
function send() {
  const nameBox = document.getElementById('nameBox');
  const textBox = document.getElementById('textBox');

  // 빈 칸이면 전송 안 함
  if (textBox.value.trim() === '') return;

  // 서버로 이름과 대화내용 전송
  socket.emit('msg', {
    name: nameBox.value,
    text: textBox.value
  });

  textBox.value = ''; // 보낸 후 입력창 비우기
}

// 2. 서버에서 메시지를 받았을 때 처리하는 함수
socket.on('msg', (data) => {
  const chatBox = document.getElementById('chatBox');
  const div = document.createElement('div');

  // 핵심: 메시지를 보낸 사람의 ID가 내 소켓 ID와 똑같은지 비교
  if (data.myId === socket.id) {
    // 나라면 오른쪽(me) 노란색 말풍선 만들기
    div.className = 'bubble-frame me';
    div.innerHTML = `<div class="text">${data.text}</div>`;
  } else {
    // 남이라면 왼쪽(others) 흰색 말풍선 만들기
    div.className = 'bubble-frame others';
    div.innerHTML = `
```

```
<div class="name">${data.name}</div>
<div class="text">${data.text}</div>
`;
  }

  chatBox.appendChild(div);
  chatBox.scrollTop = chatBox.scrollHeight; // 스크롤을 맨 아래로 이동
});
</script>

</body>
</html>
```

코드설명

```
const socket = io();
```

코드설명

```
socket.emit('msg', {  
  name: nameBox.value,  
  text: textBox.value  
});
```

```
textBox.value = '';  
}
```

코드설명

```
socket.on('msg', (data) => {
```

코드설명

```
if (data.myId === socket.id)
```

코드설명

```
div.className = 'bubble-frame me';
```

```
div.className = 'bubble-frame others';
```

코드설명

```
chatBox.appendChild(div);  
chatBox.scrollTop = chatBox.scrollHeight
```

전체 코드

```
1  const express = require('express');
2  const http = require('http');
3  const { Server } = require('socket.io');
4  const path = require('path');
5
6  const app = express();
7  const server = http.createServer(app);
8  const io = new Server(server);
9
10 app.get('/', (req, res) => {
11   res.sendFile(path.join(__dirname, 'index.html'));
12 });
13
14 io.on('connection', (socket) => {
15   socket.on('msg', (data) => {
16     io.emit('msg', {
17       myId: socket.id,
18       name: data.name,
19       text: data.text
20     });
21   });
22 });
23
24 server.listen(3000, () => {
25   console.log('서버 실행 중: http://localhost:3000');
26 });
```

코드 설명

```
const app = express();  
const server = http.createServer(app);  
const io = new Server(server);
```

코드 설명

```
app.get('/', (req, res) => {  
  res.sendFile(path.join(__dirname, 'index.html'));  
});
```

코드 설명

```
io.on('connection', (socket) => {  
  socket.on('msg', (data) => {  
    io.emit('msg', {  
      myId: socket.id,  
      name: data.name,  
      text: data.text  
    });  
  });  
});
```

코드 설명

```
server.listen(3000, () => {  
  console.log('서버 실행 중: http://localhost:3000');  
});
```

시연

```
PS C:\Users\LG\ch-app> node server.js  
서버 실행 중: http://localhost:3000
```

시연

```
PS C:\Users\LG\ch-app> node server.js  
서버 실행 중: http://localhost:3000
```



시연

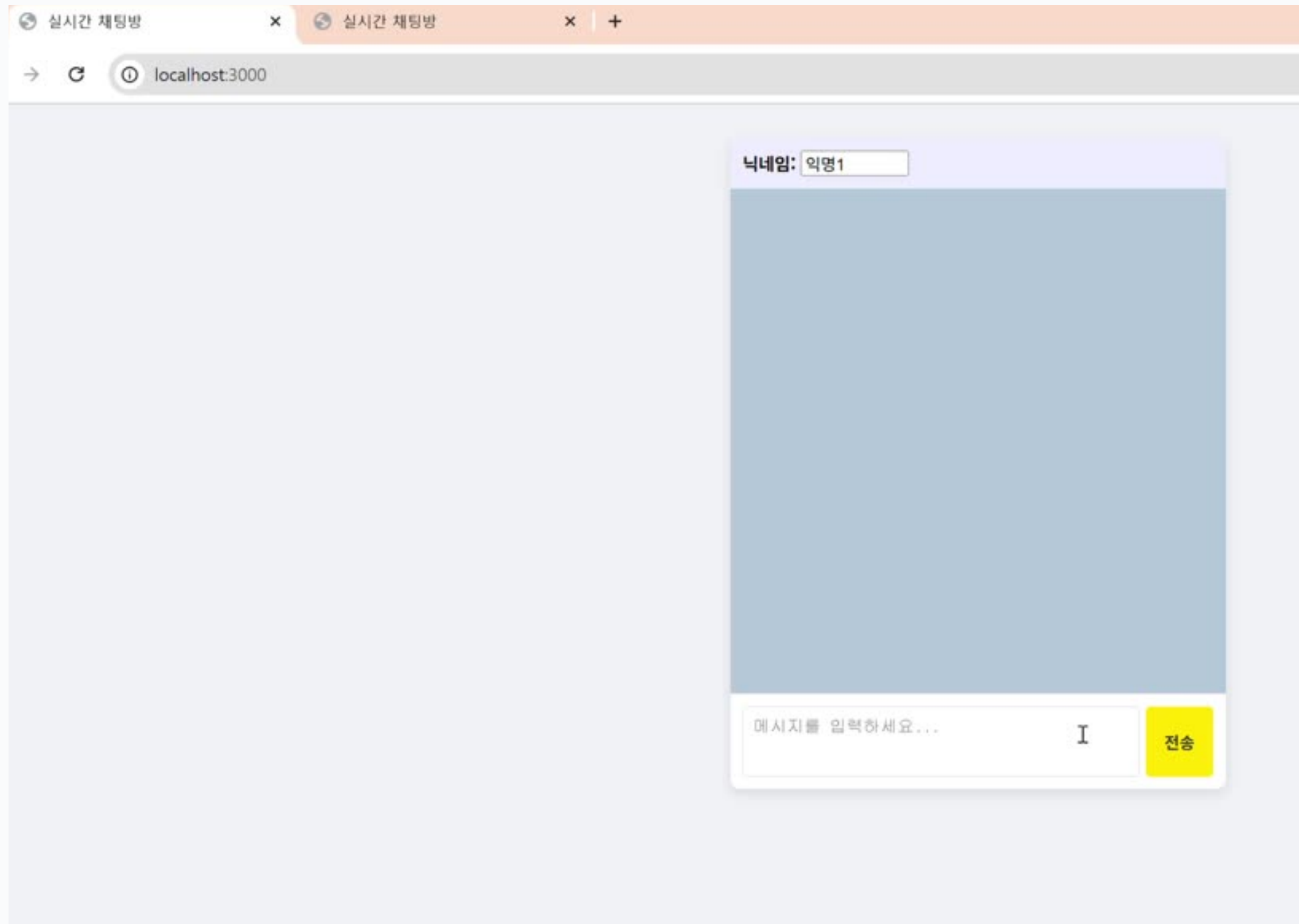
닉네임:

익명

메시지를 입력하세요...

전송

시연



시연

```
PS C:\Users\LG\ch-app>
```

시연



사이트에 연결할 수 없음

localhost에서 연결을 거부했습니다.

다음 방법을 시도해 보세요.

- 연결 확인
- 프록시 및 방화벽 확인

ERR_CONNECTION_REFUSED

새로고침

세부정보

소감



발표를 들어주셔서
감사합니다