

# Socket Programming

This project creates a real-time drawing app using HTML, JavaScript, Node.js, and Socket.io for WebSocket communication. The HTML file sets up a canvas where users can draw. JavaScript listens for mouse events on the canvas, allowing users to draw lines. Drawing events are sent to the server via Socket.io, which then broadcasts the drawing data to all connected clients, enabling real-time collaboration. The Node.js server handles incoming connections and broadcasting of drawing data to synchronize the drawing canvas across all clients.

## Index.html

```
<!DOCTYPE html>
<html Lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Real-Time Drawing App</title>
  <style>
    #canvas {
      border: 1px solid black;
    }
  </style>
</head>
<body>
  <canvas id="canvas" width="800" height="600"></canvas>

  <script
src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.1.3/socket.io.js"></scrip
t>
  <script>
    var socket = io.connect('http://192.168.1.25:3000');
    var canvas = document.getElementById('canvas');
    var context = canvas.getContext('2d');

    var drawing = false;
    var lastX, lastY;

    canvas.addEventListener('mousedown', function(e) {
      drawing = true;
```

```
[lastX, lastY] = [e.offsetX, e.offsetY];
});

canvas.addEventListener('mousemove', function(e) {
  if (!drawing) return;

  var currentX = e.offsetX;
  var currentY = e.offsetY;

  drawLine(lastX, lastY, currentX, currentY);

  socket.emit('draw', { lastX, lastY, currentX, currentY });

  lastX = currentX;
  lastY = currentY;
});

canvas.addEventListener('mouseup', function() {
  drawing = false;
});

function drawLine(x1, y1, x2, y2) {
  context.beginPath();
  context.moveTo(x1, y1);
  context.lineTo(x2, y2);
  context.stroke();
}

socket.on('draw', function(data) {
  drawLine(data.lastX, data.lastY, data.currentX, data.currentY);
});
</script>
</body>
</html>
```

## App.js

```
// server.js
const http = require('http');
const socketIo = require('socket.io');

// Create an HTTP server
const server = http.createServer((req, res) => {
  res.end('Server is running');
});

const io = socketIo(server, {
  cors: {
    origin: '*',
    methods: ['GET']
  }
});

io.on('connection', (socket) => {
  console.log('A user connected');

  socket.on('draw', (data) => {
    socket.broadcast.emit('draw', data); // Broadcast drawing data to all other
clients except the sender
  });

  socket.on('disconnect', () => {
    console.log('User disconnected');
  });
});

// Start the server
const PORT = process.env.PORT || 3000;
server.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

```
"dependencies": {
  "socket.io": "^4.7.4"
}
```

