

Ws in Node.js (Code)

- Initialize an empty Node.js project

```
npm init -y
```

- Add tsconfig to it

```
npm install typescript  
npx tsc --init
```

- Update tsconfig

```
"rootDir": "./src",  
"outDir": "./dist",
```

- Install ws

```
npm i ws @types/ws
```

Code using http library

```
import WebSocket, { WebSocketServer } from 'ws';  
import http from 'http';  
  
const server = http.createServer(function(request: any, response: any) {  
  console.log((new Date()) + ' Received request for ' + request.url);  
  response.end("hi there");  
});
```

```

const wss = new WebSocketServer({ server });

wss.on('connection', function connection(ws) {
  ws.on('error', console.error);

  ws.on('message', function message(data, isBinary) {
    wss.clients.forEach(function each(client) {
      if (client.readyState === WebSocket.OPEN) {
        client.send(data, { binary: isBinary });
      }
    });
  });

  ws.send('Hello! Message From Server!!');
});

server.listen(8080, function() {
  console.log((new Date()) + ' Server is listening on port 8080');
});

```

Code using express

```
npm install express @types/express
```

```

import express from 'express'
import { WebSocketServer } from 'ws'

const app = express()
const httpServer = app.listen(8080)

const wss = new WebSocketServer({ server: httpServer });

```

```

wss.on('connection', function connection(ws) {
  ws.on('error', console.error);

  ws.on('message', function message(data, isBinary) {
    wss.clients.forEach(function each(client) {
      if (client.readyState === WebSocket.OPEN) {
        client.send(data, { binary: isBinary });
      }
    });
  });

  ws.send('Hello! Message From Server!!');
});

```

Code without HTTP Servers

```

import WebSocket, { WebSocketServer } from 'ws';

const wss = new WebSocketServer({ port: 8080 });

wss.on('connection', function connection(ws) {
  ws.on('error', console.error);

  ws.on('message', function message(data, isBinary) {
    wss.clients.forEach(function each(client) {
      if (client.readyState === WebSocket.OPEN) {
        client.send(data, { binary: isBinary });
      }
    });
  });
});

```