**1. TCP Server**

The server listens for incoming client connections on a specific port and processes the data sent by the clients.

Here’s how you can create a simple TCP server in Node.js:

javascript

Copy code

// Import the 'net' module

const net = require('net');

// Create a TCP server

const server = net.createServer((socket) => {

console.log('Client connected');

// Handle incoming data from the client

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

console.log(`Received from client: ${data}`);

// Send a response back to the client

socket.write('Hello from the server!');

});

// Handle when the client closes the connection

socket.on('end', () => {

console.log('Client disconnected');

});

// Handle errors

socket.on('error', (err) => {

console.error(`Socket error: ${err.message}`);

});

});

// The server listens on port 4000

server.listen(4000, () => {

console.log('Server listening on port 4000');

});

**Explanation:**

* **net.createServer()**: Creates a TCP server.
* **socket.on('data', ...)**: Handles incoming data from the client.
* **socket.write()**: Sends data back to the client.
* **server.listen(4000)**: The server listens on port 4000 for incoming connections.

**2. TCP Client**

The client connects to the server on a specific port and sends data. Here’s how you can create a TCP client:

javascript

Copy code

// Import the 'net' module

const net = require('net');

// Create a TCP client and connect to the server

const client = net.createConnection({ port: 4000 }, () => {

console.log('Connected to the server');

// Send data to the server

client.write('Hello from the client!');

});

// Handle data received from the server

client.on('data', (data) => {

console.log(`Received from server: ${data}`);

// Close the connection after receiving the data

client.end();

});

// Handle when the connection is closed

client.on('end', () => {

console.log('Disconnected from the server');

});

// Handle errors

client.on('error', (err) => {

console.error(`Client error: ${err.message}`);

});

### ****2. TCP Client****

The client connects to the server on a specific port and sends data. Here’s how you can create a TCP client:

javascript

Copy code

// Import the 'net' module

const net = require('net');

// Create a TCP client and connect to the server

const client = net.createConnection({ port: 4000 }, () => {

console.log('Connected to the server');

// Send data to the server

client.write('Hello from the client!');

});

// Handle data received from the server

client.on('data', (data) => {

console.log(`Received from server: ${data}`);

// Close the connection after receiving the data

client.end();

});

// Handle when the connection is closed

client.on('end', () => {

console.log('Disconnected from the server');

});

// Handle errors

client.on('error', (err) => {

console.error(`Client error: ${err.message}`);

});