FEJLESZTŐI DOKUMENTÁCIÓ

Készítette: Herczeg Ferenc

Követelményspecifikáció

A projekt célja egy olyan szerver és kliens alkalmazás létrehozása, amelyben a szerver és a kliens egymással üzenet formájában képes kommunikálni a terminálon keresztül.

Szerver:

Létrejön egy szerver egy meghatározott porton, és addig várakozik, amíg egy kliens nem csatlakozik. Amennyiben ez megtörténik, a szerver képes a terminálon keresztül üzenetet küldeni a kliensnek, és ugyanúgy üzenetet is fogadni.

Kliens:

A kliens képes a meghatározott porton keresztül a szerverhez kapcsolódni, és ezáltal a szerver részére a terminálon keresztül üzenetet küldeni, valamint a szervertől is üzenetet fogadni.

Környezet

Mind a szerver és mind a kliens is Javascript nyelven íródott, azon belül is a Node.js szoftverrendszer segítségével. A projekt a v14.16.1. Node.js verzióval készült. Ezen kívül a kódok a Visual Studio Code nevű programban lett írva, és annak a belső Termináljával történt a tesztelés.

Szerver kódja

1. Net modul

A szerver a net modul betöltésével kezdődik.

var net = require('net');

1. Betűszín

<https://www.npmjs.com/package/chalk>

Ezen az oldalon található npm chalk modul segítségével van lehetőség a terminal-ra beírt szöveg színét változtatni. Ehhez a terminal segítségével telepíteni kell a package-t (Install):

$ npm install chalk

Ezt követően az Usage-ben látható a használata:

const chalk = require('chalk');

console.log(chalk.blue('Hello world!'));

A chalk konstans változóra hivatkozva lehet az egyes kiíratások színét megváltoztatni.

1. User input

<https://www.codecademy.com/articles/getting-user-input-in-node-js>

Erre azért van szükség, mert a terv szerint a Szerver és a Kliens is meg fog adni egy nevet, és az azt követő interakciók már azokkal a nevekkel fog megtörténni. Ennek a package-nek a működéséhez az alábbi parancs lefuttatása szükséges a terminalon:

$ npm install prompt-sync

Működése:

const prompt = require('prompt-sync')();  
   
const name = prompt('What is your name?');  
console.log(`Hey there ${name}`);

Ugyanezzel a módszerrel lett a kódban is implementálva azzal a különbséggel, hogy az üdvözlő üzenet a chalk modullal zöld színűre lett állítva. A parancs megkérdezi a nevet (ékezetet nem érzékel), és köszön zöld színnel, majd ezt a nevet fogja kiírni az üzeneteknél

const prompt = require('prompt-sync')();

const name = prompt('Mi a neved? ');

console.log(chalk.green('Üdv, ' + name + '!'));

1. Port és szerver létrehozása

let port = 3000;

var server = net.createServer(

1. Függvény létrehozása, amelyben a kiíratások meg fognak valósulni (az elkövetkezőkben a client paraméterrel fog megtörténni a klienssel történő interakció)

function (client){

1. Kliens port és cím kiírása zöld színnel

console.log(chalk.green('Egy kliens csatlakozott a ' + client.remotePort + ' porton, és a ' + client.remoteAddress + ' címen.'));

console.log(chalk.green('Kezdhetitek a beszélgetést!'))

1. Szerverhez beírt üzenet (buffer) a szerver által megadott név (name) alapján a kliensnél cián színnel történő kiíratása

process.stdin.on('data',

function(buffer){

client.write(chalk.cyan(name + ' üzenete: ' + buffer));

}

);

1. Klienstől kapott üzenet (buffer) kiírása a szerveren sárgával

client.on('data',

function(buffer){

process.stdout.write(chalk.yellow(buffer));

}

);

}

);

1. Szerver várakozik a 3000-es porton

console.log('A szerver várakozik a ' + port + '-es porton!')

server.listen(port);

Kliens kódja

1. Net modul

A szerver a net modul betöltésével kezdődik.

var net = require('net');

1. Betűszín

<https://www.npmjs.com/package/chalk>

Ezen az oldalon található npm chalk modul segítségével van lehetőség a terminal-ra beírt szöveg színét változtatni. Ehhez a terminal segítségével telepíteni kell a package-t (Install):

$ npm install chalk

Ezt követően az Usage-ben látható a használata:

const chalk = require('chalk');

console.log(chalk.blue('Hello world!'));

A chalk konstans változóra hivatkozva lehet az egyes kiíratások színét megváltoztatni.

1. User input

<https://www.codecademy.com/articles/getting-user-input-in-node-js>

Erre azért van szükség, mert a terv szerint a Szerver és a Kliens is meg fog adni egy nevet, és az azt követő interakciók már azokkal a nevekkel fog megtörténni. Ennek a package-nek a működéséhez az alábbi parancs lefuttatása szükséges a terminalon:

$ npm install prompt-sync

Működése:

const prompt = require('prompt-sync')();  
   
const name = prompt('What is your name?');  
console.log(`Hey there ${name}`);

Ugyanezzel a módszerrel lett a kódban is implementálva azzal a különbséggel, hogy az üdvözlő üzenet a chalk modullal zöld színűre lett állítva. A parancs megkérdezi a nevet (ékezetet nem érzékel), és köszön zöld színnel, majd ezt a nevet fogja kiírni az üzeneteknél

const prompt = require('prompt-sync')();

const name = prompt('Mi a neved? ');

console.log(chalk.green('Üdv, ' + name + '!'));

1. Port és Socket létrehozása

let port = 3000;

var sock = net.Socket();

1. Csatlakozás a szerverhez

sock.connect(port, "localhost");

sock.on('connect',

function(){

1. Sikeres csatlakozás esetén üzenet kiíratás zöld színnel

console.log(chalk.green('Szerverhez csatlakozva!'));

console.log(chalk.green('Kezdhetitek a beszélgetést!'))

1. Klienshez beírt üzenet (buffer) a kliens által megadott név (name) alapján üzenet kiírása a szervernél

process.stdin.on('data',

function(buffer){

sock.write(name + ' üzenete: ' + buffer);

}

);

}

);

1. Szervertől kapott üzenet (buffer) kiírása a kliensen

sock.on('data',

function(buffer){

process.stdout.write(buffer);

}

);

Nehézségek

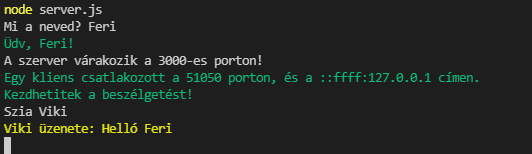
A jelenlegi működés az abból áll, hogy ha például a szerver beír egy üzenetet, akkor a saját terminálján csak simán fehérrel, név nélkül íródik ki az üzenet, és a kliensnél íródik ki a megfelelő színnel, valamint névvel az üzenet. Ugyanez igaz fordítva is, ha a klienshez írunk be egy üzenetet.

Az eredeti elképzelés szerint, ha például a szerver beír egy üzenetet, akkor már a saját terminálra is a megváltozott színnel, valamint ezzel a szöveggel: „A te üzeneted: ” kellett volna megvalósulnia, de ezt még nem sikerült kivitelezni.

Tesztelés

A tesztelés során a szerver nevének „Feri” lesz adva, míg a kliensnek „Viki”. Miután a két név beírása megtörtént a két oldalon, mindkét terminálba egy köszönő üzenet lesz beírva.

Szerver:



Kliens:

