Multiplayer Game with Unity and Node js – Part #1- The Setup and Developing a Connection

1. Download node.js at this link (the LTS version) [https://nodejs.org/en/](https://www.youtube.com/redirect?v=J0udhTJwR88&redir_token=pPLHIhcJS063jHEENLeF2I79sgF8MTU4Mzc4ODgwNEAxNTgzNzAyNDA0&q=https%3A%2F%2Fnodejs.org%2Fen%2F&event=video_description)

Download visual studio too. [https://code.visualstudio.com/](https://www.youtube.com/redirect?v=J0udhTJwR88&redir_token=pPLHIhcJS063jHEENLeF2I79sgF8MTU4Mzc4ODgwNEAxNTgzNzAyNDA0&q=https%3A%2F%2Fcode.visualstudio.com%2F&event=video_description)

Open visual studio. Make a new folder. Call it “Server”.

Now open a terminal (view->integrated terminal)

Type:node -v

To get version. Now we must integrate our node file we just downloaded.

Type:npm init

Package Name:gameserver

Entry Point: index.js

Authour: Your name

And remember to reference to this json file for your build each time.

Now type:npm install nodemon

Let it install.

Now type:npm install socket.io

Now type:npm install shorted

Now its time to create a new file. Call it “Index.js” in the visual studio.

1. Open Unity and Create a new project. Call it what you would like to call your game.

Now open the asset store in unity (at the top) Search “Socket.IO” search the one done by Fabio. Hit import. Create your folders for your code, and inside code call it networking. Create an empty object in the game scene, call it “Code- Networking” (in the Hierarchy.).

Make a new script inside networking. Call It “NetworkClient”

using System.Collections

using System.Collections.Generic;

using UnityEngine;

using SocketIO;

namespace Project.Networking{

public class NetworkClient : SocketIOComponent{

public override void Start(){

base.Start();

}

public override void Update(){

base.Update();

}

}

Hit f12 when hovering over SocketIOComponent

Under Public Properties, create a header attribute at the top

[Header(“Socket IO Componet”)]

Now go into Unity Interface, chance public void Start to public virtual void Start();

Save now, and add NetworkClient script to the empty object we created earlier (by clicking on the right side after clicking the empty object and click at very bottom where its adding something to the object).

Change the port after 127.0.0.1/52300🡨 via the url in the script.

1. Now it is time to update index.js in your virtual studio, and time to make a connection to the server. This group of functions require socket.io, pass functions into socket.io. This is perfect for making connection.

var io = require(‘socket.io’)(process\_env\_PORT || 52300);

console.log(‘Server has started’);

io.on(‘connection’, function(socket) {

console.log(‘Connection made!’);

socket.on(‘disconnect’, function() {

console.log(‘A player has disconnected’);

});

});

1. Time to update the functions in unity so we can make a connection via debug logs on the server to test actions made to the server. This can be used with mySQL to conversate with the server at the same time

using System.Collections

using System.Collections.Generic;

using UnityEngine;

using SocketIO;

namespace Project.Networking{

public class NetworkClient : SocketIOComponent{

public override void Start(){

base.Start();

setupEvents();

}

public override void Update(){

base.Update();

}

private void setupEvents() {

On(“open”, (E) -> {

Debug.Log(“Connection made to the server”);

});

}

}

}