

Forum:[Hack Forums](#)

Thread: [Multi-client TCP Socket Server | C#](#)

itsnode - October 10, 2017, 5:58 pm

```
Code using System;
using System.Collections.Generic;
using System.Net;
using System.Net.Sockets;
using System.Text;

namespace MultiServer
{
    class Program
    {
        private static readonly Socket serverSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        private static readonly List<Socket> clientSockets = new List<Socket>();
        private const int BUFFER_SIZE = 2048;
        private const int PORT = 100;
        private static readonly byte[] buffer = new byte[BUFFER_SIZE];

        static void Main()
        {
            Console.Title = "Server";
            SetupServer();
            Console.ReadLine(); // When we press enter close everything
            CloseAllSockets();
        }

        private static void SetupServer()
        {
            Console.WriteLine("Setting up server...");
            serverSocket.Bind(new IPEndPoint(IPAddress.Any, PORT));
            serverSocket.Listen(0);
            serverSocket.BeginAccept(AcceptCallback, null);
            Console.WriteLine("Server setup complete");
        }

        /// <summary>
        /// Close all connected client (we do not need to shutdown the server socket as its connections
        /// are already closed with the clients).
        /// </summary>
        private static void CloseAllSockets()
        {
            foreach (Socket socket in clientSockets)
            {
                socket.Shutdown(SocketShutdown.Both);
                socket.Close();
            }
        }
    }
}
```

```

        serverSocket.Close();
    }

    private static void AcceptCallback(IAsyncResult AR)
    {
        Socket socket;

        try
        {
            socket = serverSocket.EndAccept(AR);
        }
        catch (ObjectDisposedException) // I cannot seem to avoid this (on exit when properly
closing sockets)
        {
            return;
        }

        clientSockets.Add(socket);
        socket.BeginReceive(buffer, 0, BUFFER_SIZE, SocketFlags.None, ReceiveCallback,
socket);
        Console.WriteLine("Client connected, waiting for request...");
        serverSocket.BeginAccept(AcceptCallback, null);
    }

    private static void ReceiveCallback(IAsyncResult AR)
    {
        Socket current = (Socket)AR.AsyncState;
        int received;

        try
        {
            received = current.EndReceive(AR);
        }
        catch (SocketException)
        {
            Console.WriteLine("Client forcefully disconnected");
            // Don't shutdown because the socket may be disposed and its disconnected anyway.
            current.Close();
            clientSockets.Remove(current);
            return;
        }

        byte[] recBuf = new byte[received];
        Array.Copy(buffer, recBuf, received);
        string text = Encoding.ASCII.GetString(recBuf);
        Console.WriteLine("Received Text: " + text);

        if (text.ToLower() == "get time") // Client requested time
        {
            Console.WriteLine("Text is a get time request");
            byte[] data = Encoding.ASCII.GetBytes(DateTime.Now.ToLongTimeString());
            current.Send(data);
        }
    }

```

```

        Console.WriteLine("Time sent to client");
    }
    else if (text.ToLower() == "exit") // Client wants to exit gracefully
    {
        // Always Shutdown before closing
        current.Shutdown(SocketShutdown.Both);
        current.Close();
        clientSockets.Remove(current);
        Console.WriteLine("Client disconnected");
        return;
    }
    else
    {
        Console.WriteLine("Text is an invalid request");
        byte[] data = Encoding.ASCII.GetBytes("Invalid request");
        current.Send(data);
        Console.WriteLine("Warning Sent");
    }
}

current.BeginReceive(buffer, 0, BUFFER_SIZE, SocketFlags.None, ReceiveCallback,
current);
}
}
}
}

```

BahNahNah - October 10, 2017, 10:17 pm

Each client should have its own buffer. Without it there is potential for two clients to write to the buffer at the same time and corrupt the data.

kevy42 - October 11, 2017, 10:24 am

(Oct 10, 2017 10:17 PM)BahNahNah Wrote: [\[CITE\]](#) Each client should have its own buffer. Without it there is potential for two clients to write to the buffer at the same time and corrupt the data.

you sir.. just solved a headache for me, thanks. iv had this issue for ages and didnt know what was causing it

0xDEAD10CC - October 11, 2017, 4:28 pm

(Oct 11, 2017 10:24 AM)WeedmasterOG Wrote: [\[CITE\]](#) you sir.. just solved a headache for me, thanks. iv had this issue for ages and didnt know what was causing it

Was it not obvious conceptually from the start? ?

This code has poor string comparison, race conditions, and more.

kevy42 - October 11, 2017, 11:01 pm

(Oct 11, 2017 04:28 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Was it not obvious conceptually from the start? ?

This code has poor string comparison, race conditions, and more.

Im usually pretty decent at C# and prob solving, but for some reason sockets are like 3x haeder for me. Iv been trying to fully learn ASYNC sockets for a good while now

ForceJew - October 12, 2017, 3:46 am

<https://github.com/AbleOpus/NetworkingSamples/tree/master/MultiServer>

This is a direct copy of this. No credit no nothing.

Honestly fuck you, you could at least tell where you got it from:)
You are worst than a copy paster skid, you don't even give credit...

Ninja - October 12, 2017, 10:06 am

(Oct 11, 2017 11:01 PM)WeedmasterOG Wrote: [\[CITE\]](#) Im usually pretty decent at C# and prob solving
Don't flatter yourself.

And I suggest making a wrapper for sockets. Abstract it so it is easier to use Here is one I made that has a timeout feature, easier to use send and receive methods, and socks protocol support:
<https://hackforums.net/showthread.php?tid=5618361>

Nick (Name) - October 12, 2017, 11:38 am

(Oct 12, 2017 10:06 AM)Ninja Wrote: [\[CITE\]](#) Don't flatter yourself.

And I suggest making a wrapper for sockets. Abstract it so it is easier to use Here is one I made that has a timeout feature, easier to use send and receive methods, and socks protocol support:
<https://hackforums.net/showthread.php?tid=5618361>
Or use DragonHunter's stuff
<https://github.com/AnguisCaptor/SecureSocketProtocol>

kevy42 - October 12, 2017, 1:32 pm

(Oct 12, 2017 10:06 AM)Ninja Wrote: [\[CITE\]](#) Don't flatter yourself.

And I suggest making a wrapper for sockets. Abstract it so it is easier to use Here is one I made that has a timeout feature, easier to use send and receive methods, and socks protocol support:
<https://hackforums.net/showthread.php?tid=5618361>
? u havent even seen most of my work, so dropping a comment like that is just pointless. But il look into the wrapper, should make things easier, thanks

(Oct 12, 2017 11:38 AM)Nick (Name) Wrote: [\[CITE\]](#) Or use DragonHunter's stuff
<https://github.com/AnguisCaptor/SecureSocketProtocol>
Will check this out too, thanks

Rhez - October 12, 2017, 11:47 pm

Sooo, is this you?
<https://github.com/AbleOpus>

Ninja - October 13, 2017, 7:46 am

(Oct 12, 2017 01:32 PM)WeedmasterOG Wrote: [\[CITE\]](#) ? u havent even seen most of my work, so dropping a comment like that is just pointless. But il look into the wrapper, should make things easier, thanks
What I have seen was absolute shit.

kevy42 - October 13, 2017, 10:07 am

(Oct 13, 2017 07:46 AM)Ninja Wrote: [\[CITE\]](#) What I have seen was absolute shit.
What?, on my github?. Iv had multiple ppl tell me that they learnt a good bit from the small projects iv got on there and some HQ ppl have also dropped some nice comments

gigajew - January 18, 2018, 7:40 pm

If you want async tasks instead of Begin and End, here's one method:
<https://codereview.stackexchange.com/questions/177526/c-async-socket-wrapper>
Code public class SimpleSocket : ISimpleSocket

```

{
    private readonly Socket _socket;

    private SimpleSocket(Socket socket)
    {
        _socket = socket;
    }

    public SimpleSocket()
    {
        _socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    }

    public void Bind(int port)
    {
        var endPoint = new IPEndPoint(IPAddress.Any, port);
        _socket.Bind(endPoint);
    }

    public void Listen(int backlog)
    {
        _socket.Listen(backlog);
    }

    public async Task<ISimpleSocket> AcceptAsync()
    {
        var socket = await Task.Factory.FromAsync(_socket.BeginAccept, _socket.EndAccept, true);
        return new SimpleSocket(socket);
    }

    public async Task ConnectAsync(string host, int port)
    {
        await Task.Factory.FromAsync(_socket.BeginConnect, _socket.EndConnect, host, port, null);
    }

    public async Task<int> ReceiveAsync(byte[] buffer, int offset, int count)
    {
        using (var stream = new NetworkStream(_socket))
        {
            return await stream.ReadAsync(buffer, offset, count);
        }
    }

    public async Task SendAsync(byte[] buffer, int offset, int count)
    {
        using (var stream = new NetworkStream(_socket))
        {
            await stream.WriteAsync(buffer, offset, count);
        }
    }

    public void Dispose()
    {

```

```

        _socket?.Dispose();
    }
}

```

Ninja - January 19, 2018, 3:09 am

(Oct 11, 2017 11:01 PM)Kevin™ Wrote: [\[CITE\]](#) Im usually pretty decent at C# and prob solving, but for some reason sockets are like 3x haeder for me. Iv been trying to fully learn ASYNC sockets for a good while now

<https://hackforums.net/showthread.php?tid=5735967>

This should help you out then.

(Jan 18, 2018 08:40 PM)gigajew Wrote: [\[CITE\]](#) If you want async tasks instead of Begin and End, here's one method:

<https://codereview.stackexchange.com/questions/177526/c-async-socket-wrapper>

```

Code public class SimpleSocket : ISimpleSocket
{
    private readonly Socket _socket;

    private SimpleSocket(Socket socket)
    {
        _socket = socket;
    }

    public SimpleSocket()
    {
        _socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    }

    public void Bind(int port)
    {
        var endPoint = new IPEndPoint(IPAddress.Any, port);
        _socket.Bind(endPoint);
    }

    public void Listen(int backlog)
    {
        _socket.Listen(backlog);
    }

    public async Task<ISimpleSocket> AcceptAsync()
    {
        var socket = await Task.Factory.FromAsync(_socket.BeginAccept, _socket.EndAccept, true);
        return new SimpleSocket(socket);
    }

    public async Task ConnectAsync(string host, int port)
    {
        await Task.Factory.FromAsync(_socket.BeginConnect, _socket.EndConnect, host, port, null);
    }

    public async Task<int> ReceiveAsync(byte[] buffer, int offset, int count)
    {
        using (var stream = new NetworkStream(_socket))

```

```

    {
        return await stream.ReadAsync(buffer, offset, count);
    }
}

public async Task SendAsync(byte[] buffer, int offset, int count)
{
    using (var stream = new NetworkStream(_socket))
    {
        await stream.WriteAsync(buffer, offset, count);
    }
}

public void Dispose()
{
    _socket?.Dispose();
}
}

```

I never thought of using this way. Thanks for the tip.

gigajew - January 19, 2018, 11:26 am

(Jan 19, 2018 04:09 AM)Ninja Wrote: [\[CITE\]](https://hackforums.net/showthread.php?tid=5735967) <https://hackforums.net/showthread.php?tid=5735967>
This should help you out then.

I never thought of using this way. Thanks for the tip.

I found a better wrapper. I can post it when I get home or you can google it

Ninja - January 19, 2018, 12:50 pm

(Jan 19, 2018 12:26 PM)gigajew Wrote: [\[CITE\]](#) I found a better wrapper. I can post it when I get home or you can google it

How can I google it if you didn't tell me anything about it? But I am curious, please PM me it.

gigajew - January 19, 2018, 1:21 pm

(Jan 19, 2018 01:50 PM)Ninja Wrote: [\[CITE\]](#) How can I google it if you didn't tell me anything about it? But I am curious, please PM me it.

"Socket wrapper await task" or similar query should pull it up

gungunfebrianza - February 3, 2018, 11:21 pm

just source no tutorial here??? better explanation will be good.