

```

if (args.SocketError == SocketError.Success)
    _listenSocket = new Socket(endPoint.AddressFamily, SocketType.Stream, Protocols.Tcp);
    _onAcceptHandler += onAcceptHandler;
    _listenSocket.Bind(endPoint);
    _listenSocket.Listen(10);

    SocketAsyncEventArgs args = new SocketAsyncEventArgs();
    args.Completed += new EventHandler<SocketAsyncEventArgs>(OnAcceptCompleted);
    RegisterAccept(args);
}

void RegisterAccept(SocketAsyncEventArgs args)
{
    args.AcceptSocket = null;

    bool pending = _listenSocket.AcceptAsync(args);
    if (pending == false)
        OnAcceptCompleted(null, args);
}

void OnAcceptCompleted(object sender, SocketAsyncEventArgs args)
{
    if (args.SocketError == SocketError.Success)
    {
        GameSession session = new GameSession();
        session.Start(args.AcceptSocket);
        session.OnConnected(args.AcceptSocket.RemoteEndPoint);
    }
    else
        Console.WriteLine(args.SocketError.ToString());
}

RegisterAccept(args);

```

```

using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace ServerCore
{
    class GameSession : Session
    {
        public override void OnConnected(EndPoint endPoint)
        {
            Console.WriteLine($"OnConnected : {endPoint}");
        }

        public override void OnDisconnected(EndPoint endPoint)
        {
            Console.WriteLine($"OnDisconnected : {endPoint}");
        }

        public override void OnRecv(ArraySegment<byte> buffer)
        {
            string recData = Encoding.UTF8.GetString(buffer.Array, buffer.Offset, buffer.Count);
            Console.WriteLine($"From Client : {recData}");
        }

        public override void OnSend(int numOfBytes)
        {
            Console.WriteLine($"Transferred bytes: {numOfBytes}");
        }
    }
}

class Program
{
    static Listener _listener = new Listener();
    static void OnAcceptHandler(Socket clientSocket)

```

① Action<Socket>
↓
② Func<Session> = SessionFactory

함수를 빼는 걸 (기억)

③ Listener.init(endpoint, func)

```

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

namespace ServerCore
{
    class Listener
    {
        Socket _listenSocket;
        Func<Session> _sessionFactory;

        public void Init(IPPEndPoint endPoint, Func<Session> sessionFactory)
        {
            _listenSocket = new Socket(endPoint.AddressFamily, SocketType.Stream, Protocols.Tcp);
            _sessionFactory = sessionFactory;

            // 문자기 교류
            _listenSocket.Bind(endPoint);

            // 영업 시작
            // backlog : 최대 대기수
            _listenSocket.Listen(10);

            SocketAsyncEventArgs args = new SocketAsyncEventArgs();
            args.Completed += new EventHandler<SocketAsyncEventArgs>(OnAcceptCompleted);
            RegisterAccept(args);
        }

        void RegisterAccept(SocketAsyncEventArgs args)
        {
            args.AcceptSocket = null;

            bool pending = _listenSocket.AcceptAsync(args);
            if (pending == false)
                OnAcceptCompleted(null, args);
        }

        void OnAcceptCompleted(object sender, SocketAsyncEventArgs args)
        {
            if (args.SocketError == SocketError.Success)
            {
                GameSession session = new GameSession();
                session.Start(args.AcceptSocket);
                session.OnConnected(args.AcceptSocket.RemoteEndPoint);
            }
            else
                Console.WriteLine(args.SocketError.ToString());
        }

        RegisterAccept(args);
    }
}

```

```

try
{
    GameSession session = new GameSession();
    session.Start(clientSocket);

    byte[] sendBuff = Encoding.UTF8.GetBytes("Welcome to MUDRG Server !");
    session.Send(sendBuff);

    Thread.Sleep(1000);

    session.Disconnect();
    session.Disconnect();
}
catch (Exception e)
{
    Console.WriteLine(e);
}

static void Main(string[] args)
{
    // DNS (Domain Name System)
    string host = Dns.GetHostName();
    IPHostEntry ipHost = Dns.GetHostEntry(host);
    IPAddress ipAddr = ipHost.AddressList[0];
    IPEndPoint endPoint = new IPEndPoint(ipAddr, 7777);

    Listener init(endPoint, () => { return new GameSession(); });

    Console.WriteLine("Listening...");

    while (true)
    {
        ;
    }
}

```

Q. ② 약 Start(args. AcceptSocket) 만

OnConnected (args. AcceptSocket. RemoteEndpoint)

혹은 네트워크에서 Client가 끊기면?

접근하기 어렵거나 예외가 발생하거나.

혹은 쌍방향 Test 해보기!

Socket, Connect → Connection을 틀 때

Why? 저사용자에게