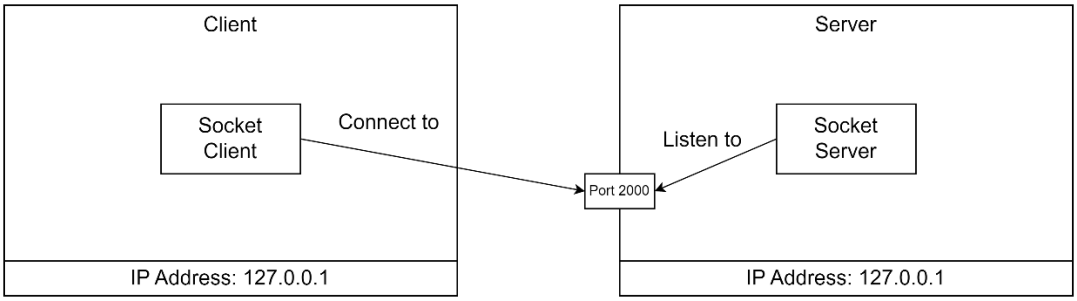
Exercise: Client – Server communication

In this exercise, you will use sockets and TCP to send data between two programs, a client and a server.



Exercise 1:
Create two .NET solutions: A client and a server. Both should be a console application.

The server shall listen for incoming connections on port 2000.
It shall print the data it receives and send a reply to the client.

The client shall connect to the server on port 2000.
It shall send a message to the server and print the reply it receives.

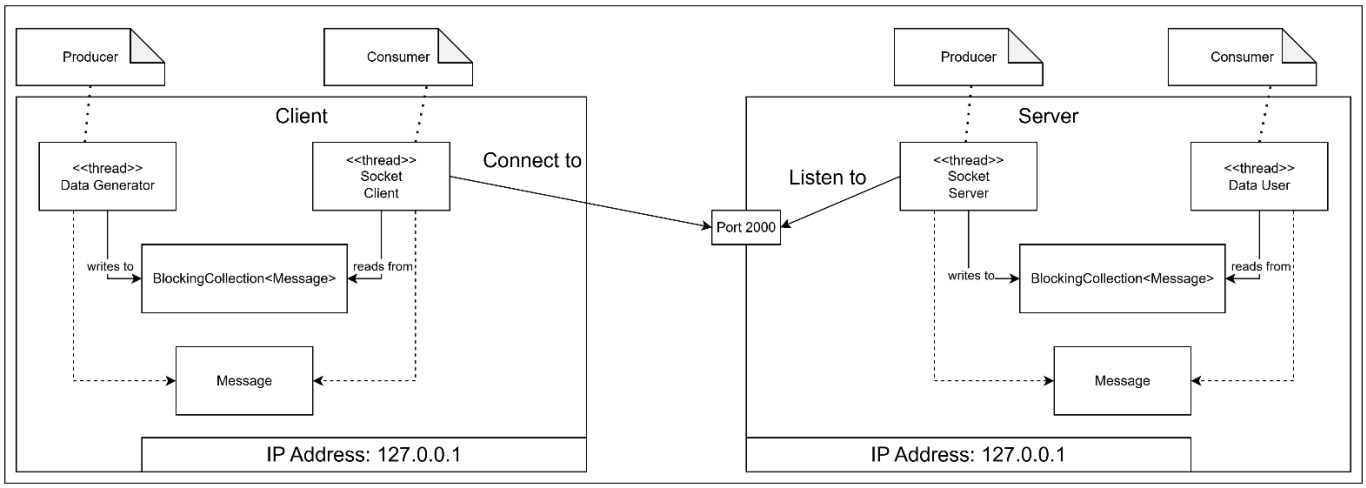
Run both client and server on the same PC and use the loopback address – 127.0.0.1

Tip: Open two Visual Studio windows and work with the client in one and the server in the other one. This way, you can start a debugging session for both client and server at the same time.

Exercise 2:
The message generation and transmission on the client side shall now be handled by two different threads and follow the producer-consumer pattern, where the producer sends data to the consumer using a queue (a BlockingCollection in C#).

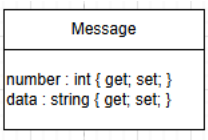
Likewise on the server side, the reception of data and processing of data shall be handled by two different threads and follow the producer-consumer pattern. The producer thread shall decode the data received on the socket, create the corresponding message and send it to the consumer using a queue (a BlockingCollection in C#).

The design will look like this:



a. Start by defining the Message class.

This shall be a simple data class (a DTO), with public properties to allow it to be used by the JSON serializer. This class must have the properties on both client and server, to allow it to be converted to/from JSON.
For a start, just give each message a number and a text string to send:



b. Now, on the client side, create the DataGenerator and send data from the DataGenerator to the SocketClient using the BlockingCollection. You can run this client without changing the server and observe, that the server now receives JSON formatted data.
c. Next, on the server, deserialize the JSON to create a Message and send it to a DataUser using the BlockingCollection.

Exercise 3:
Run the client and server on different computers on the network.
To do this, you must figure out which IP address to connect to from the client and how to listen for external connections on the server.