

# Exercise: Math server

In this exercise you must program a simple Math TCP server. The server receives requests like "ADD 4.4 5.5" and send a response with the result ("9.9" in this case).

The structure of the server is very similar to the Echo TCP server. However, the service given by the Math server is slightly more advanced than the service given by the Echo server.

## Protocol design

Before you start programming you must define a protocol: A set of rules explaining the structure of requests and responses.

I suggest a single-line request protocol:

- ADD 4.4 5.5
- SUB 6.2 2.1
- etc.

The response could be a simple number.

But you can chose another protocol if you want to, for example a multi-line protocol.

Error handling, etc.: Don't worry about this now ... it comes later in this exercise.

## Server program

Make sure you can read and understand the Echo TCP server.

Make a copy of the Visual Studio Project holding the Echo TCP server (do not modify the original project).

With the Math TCP server you must rework the part of the server where it gives service to the client.

1. Read the request // like Echo
2. Split the request into parts
3. Find the operation (ADD, SUB, etc.)  
string.Split(separator) might be useful
4. Find the numbers  
Double.Parse(numberString) might be useful
5. Perform the operation on the numbers
6. Send the response back to the client. // like Echo

## Client program

Make a simple client program (preferably Console Application in another Visual Studio solution) to try the Math TCP server.

The client program should be very similar to the Echo TCP client program.

## Testing

Use your unit testing framework to test the Math TCP server.

Eventhough this is not exactly unit testing, the unit testing framework is still useful.

The test assumes that the server is running on a known host + port.

The test send a request to the server (very much like the client program) and make assertions about the response.

In the test you can make helper (not *TestMethod*) method like

- private static String GetResponse(String request) { ... }  
This method must connect to the server, send the request, and get the response.

## **Error handling: Bad request**

The client may send a bad request, i.e. a request not according to the protocol.

Examples:

- add 4.4 5.5
- plop 4.4 5.5
- ADD 4.4 5a5

In any case the server *must* send back a proper response for example

- ERROR Illegal operation: plop
- ERROR Not a number

You will have to adapt the response part of your protocol AND add more lines to your server.

## **Trying and testing**

- Try it from the client application.
- Test it: Add more lines to you test.