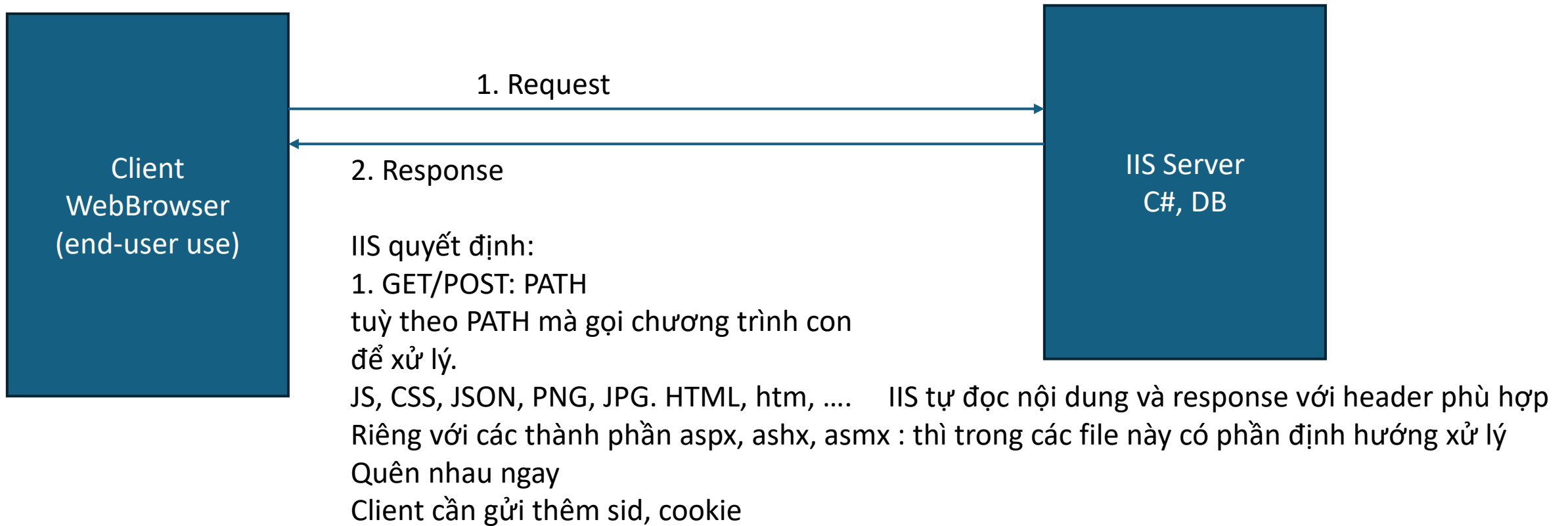
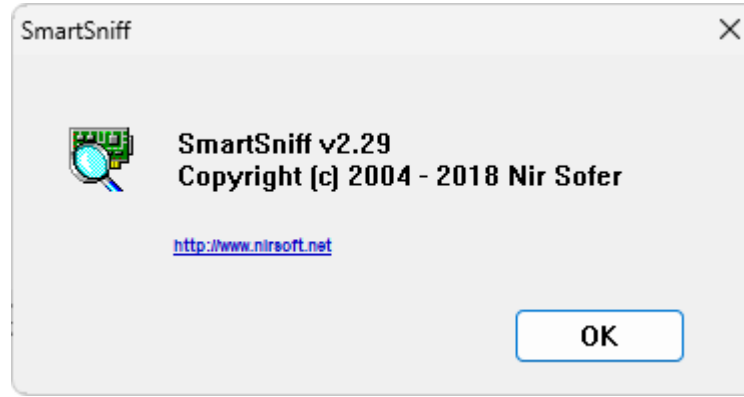


WebSocket



Sử dụng WebSocket










- Trình duyệt hiện đại thời 202x: hỗ trợ
- WebSocket: là 1 kết nối TCP tới máy chủ
- Và duy trì kết nối: 2 chiều truyền thông
- Client gửi y/c lên server
- Server phản hồi về
- Server có thể tự gửi dữ liệu xuống client mà ko cần request

Quy trình 3phuts

1. New project loại ASP.NET Web Application (.NET Framework)
2. Đặt tên cho project, chọn nơi lưu, chọn version of .NET Framework=2.0
3. Bỏ check: https (vì iis express sẽ chạy khá chậm với https ảo)
4. Edit web.config (xem hình)
5. Chuyển project về loại .NET Framework version 4.5.2 trở lên
6. Thêm thư viện hỗ trợ websocket vào project
Ínşţǎll Rắçlắgê Nĩçsộşộğţ Wêçşộçlêţş
7. Add file ws.ashx rồi edit nó (xem github để view full code): ws backend ok
8. Html+js of front-end: cần làm như sau (xem index.html trên github)

Add a new project

Recent project templates

-  ASP.NET Web Application (.NET Framework) C#
-  ASP.NET Core Web API C#
-  Class Library (.NET Framework) C#
-  Windows Forms App (.NET Framework) C#
-  Console App (.NET Framework) C#
-  Console App C#
-  Windows Forms Control Library (.NET Framework) C#
-  Flask Web Project Python
-  Web Project Python

Search for templates (Alt+S)



[Clear all](#)

C#

Windows

Web



ASP.NET Web Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

C#

Windows

Cloud

Web



Web Driver Test for Edge (.NET Core)

A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft WebDriver).

C#

Windows

Web

Test



Web Driver Test for Edge (.NET Framework)

A project that contains unit tests that can automate UI testing of web sites within Edge browser (using Microsoft WebDriver).

C#

Windows

Web

Test



WCF Service

A Web site for creating WCF services. This template does not produce a project file and has limited MSBuild support.

C#

Windows

Web

Service



WCF Service Application

[Next](#)

Configure your new project

ASP.NET Web Application (.NET Framework)

C#

Windows

Cloud

Web

Project name

demo_ws

Location

D:\my documents\Visual Studio 2022\Projects\test_websocket

...

Framework

.NET Framework 2.0

Project will be created in "D:\my documents\Visual Studio 2022\Projects\test_websocket\demo_ws\"

Back

Create

Create a new ASP.NET Web Application



Empty

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Authentication

None

Add folders & core references

☐ Web Forms

☐ MVC

☐ Web API

Advanced

☐ Configure for HTTPS

[Back](#)

[Create](#)

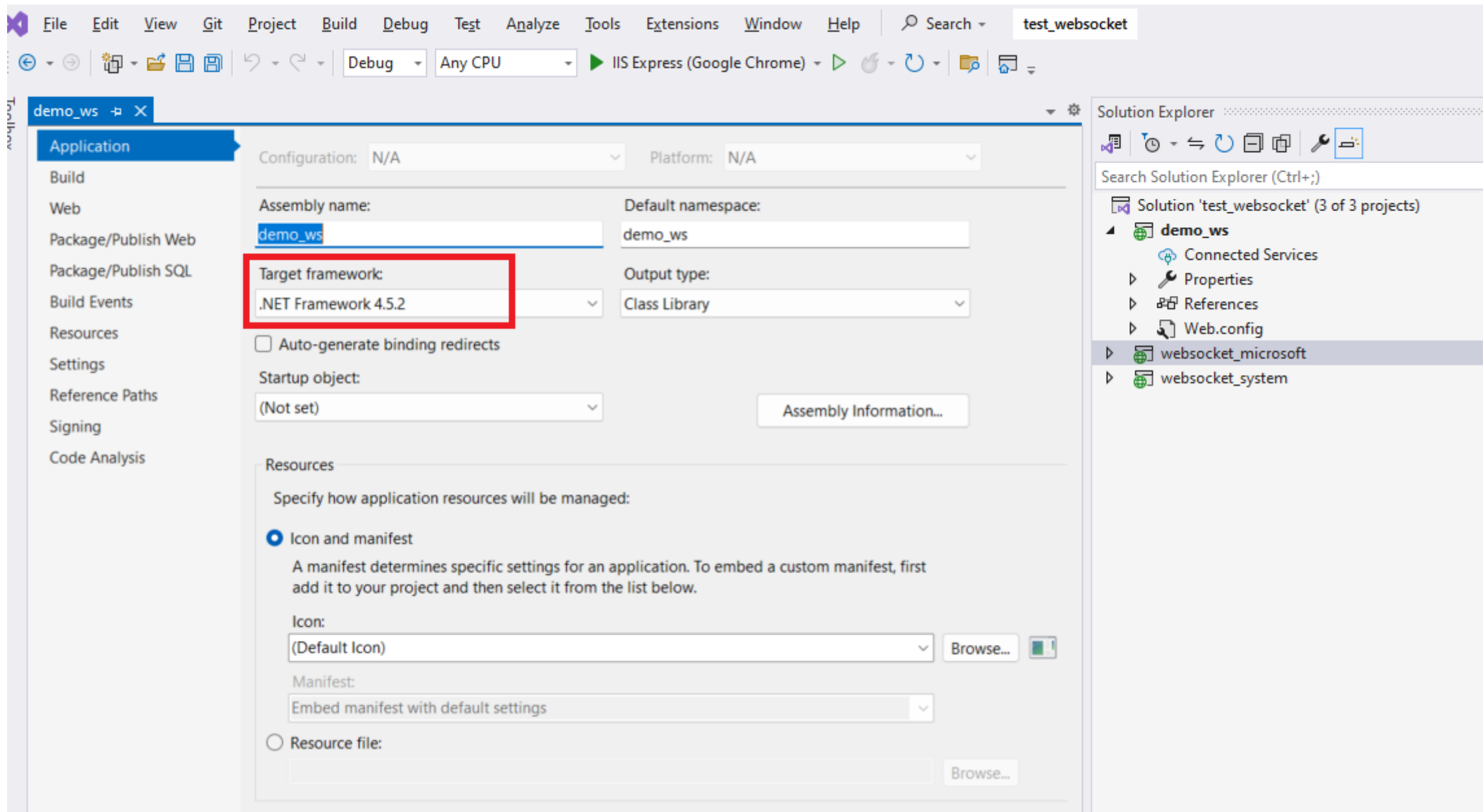
Visual Studio interface showing the Web.config file in the test_websocket solution.

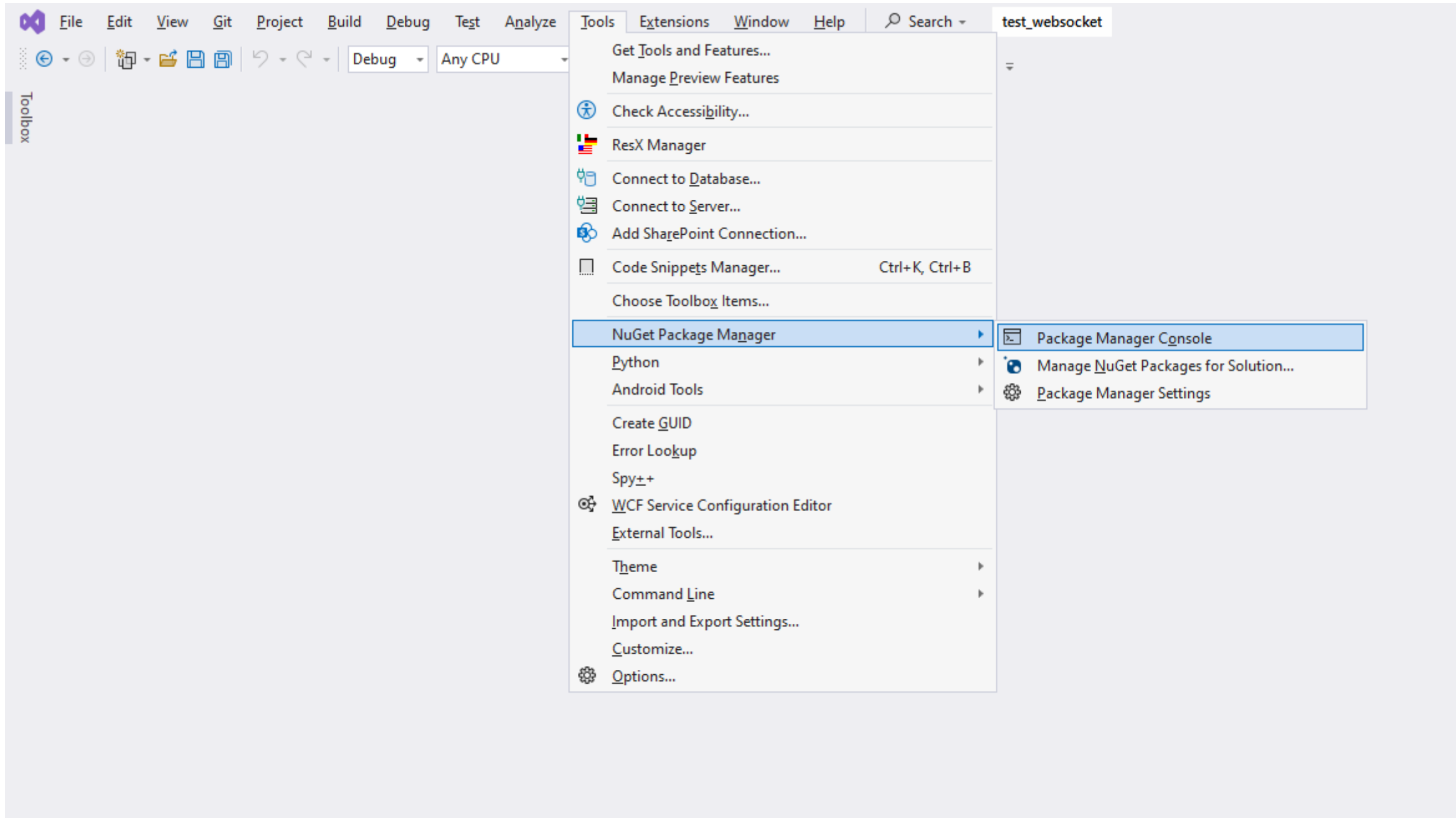
Web.config

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <httpRuntime targetFramework="4.5" />
    <compilation debug="true" />
  </system.web>
</configuration>
```

Solution Explorer

- Solution 'test_websocket' (3 of 3 projects)
- demo_ws
 - Connected Services
 - Properties
 - References
 - Web.config
- websocket_microsoft
- websocket_system





Package Manager Console

Package source: All



Default project: demo_ws



source (feed) URL to determine any dependencies.

Package Manager Console Host Version 6.11.1.2

Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package Microsoft.WebSockets

Package Manager Console

Package source: All



Default project: demo_ws



documents\Visual Studio 2022\Projects\test_websocket\packages'

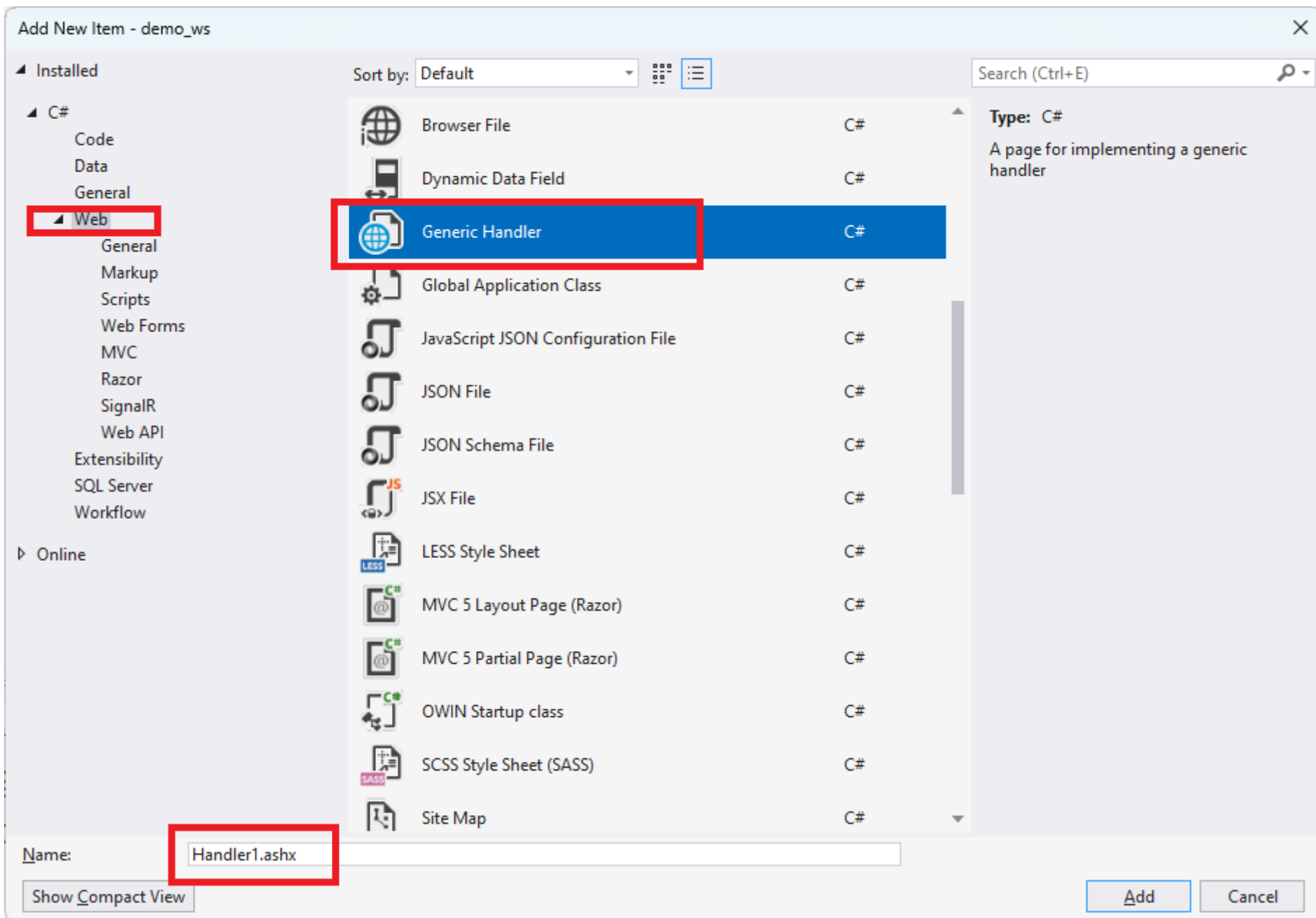
Added package 'Microsoft.WebSockets.0.2.3.1' to 'packages.config'

Successfully installed 'Microsoft.WebSockets 0.2.3.1' to demo_ws

Executing nuget actions took 1.09 sec

Time Elapsed: 00:00:03.1344317

PM> |



Visual Studio interface showing the code for `ws.ashx.cs` in the `demo_ws` project.

The code defines a `ws` class implementing `IHttpHandler` within the `demo_ws` namespace. The `ProcessRequest` method sets the response content type to `text/plain` and writes "Hello World". The `IsReusable` property returns `false`.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5
6 namespace demo_ws
7 {
8     /// <summary>
9     /// Summary description for ws
10    /// </summary>
11    public class ws : IHttpHandler
12    {
13
14        public void ProcessRequest(HttpContext context)
15        {
16            //code của ảnh chụp này tự có sau khi add 'ws.ashx'
17            context.Response.ContentType = "text/plain";
18            context.Response.Write("Hello World");
19        }
20
21        public bool IsReusable
22        {
23            get
24            {
25                return false;
26            }
27        }
28    }
29 }
```

The Solution Explorer on the right shows the project structure for `test_websocket`, including `demo_ws` and its sub-projects `websocket_microsoft` and `websocket_system`.

