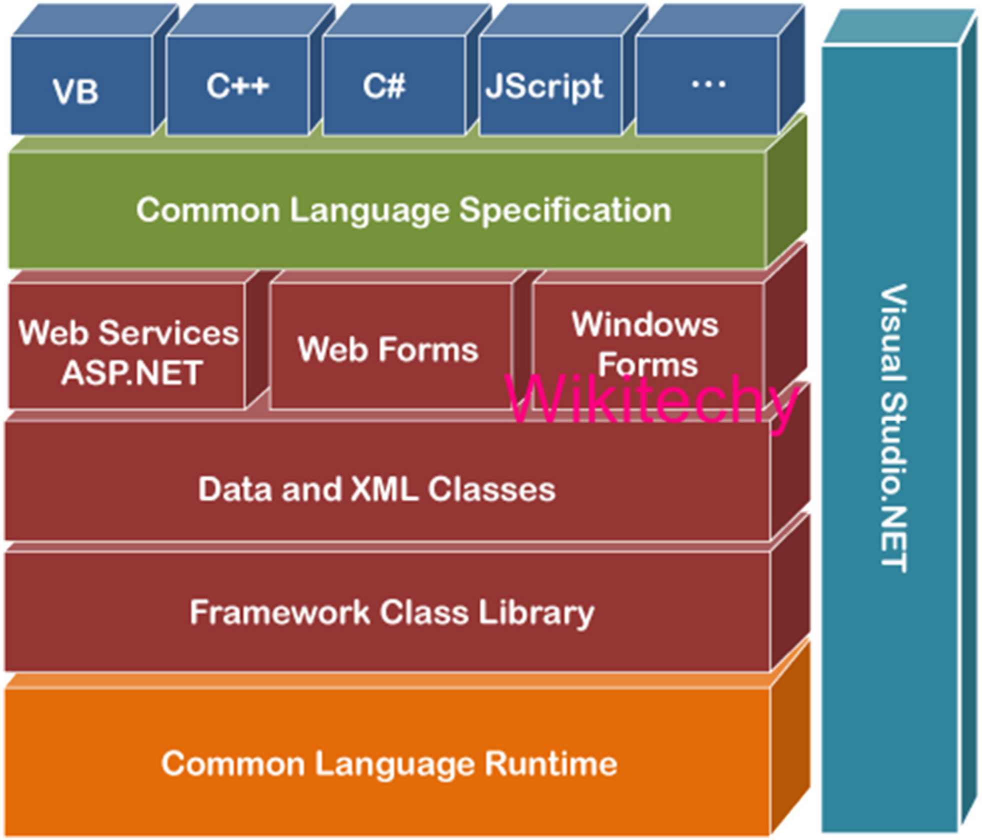
MCSE 541: Web Computing and Mining

.NET and ASP.NET Framework

Prof. Dr. Shamim Akhter
Professor, CSE, Stamford University Bangladesh

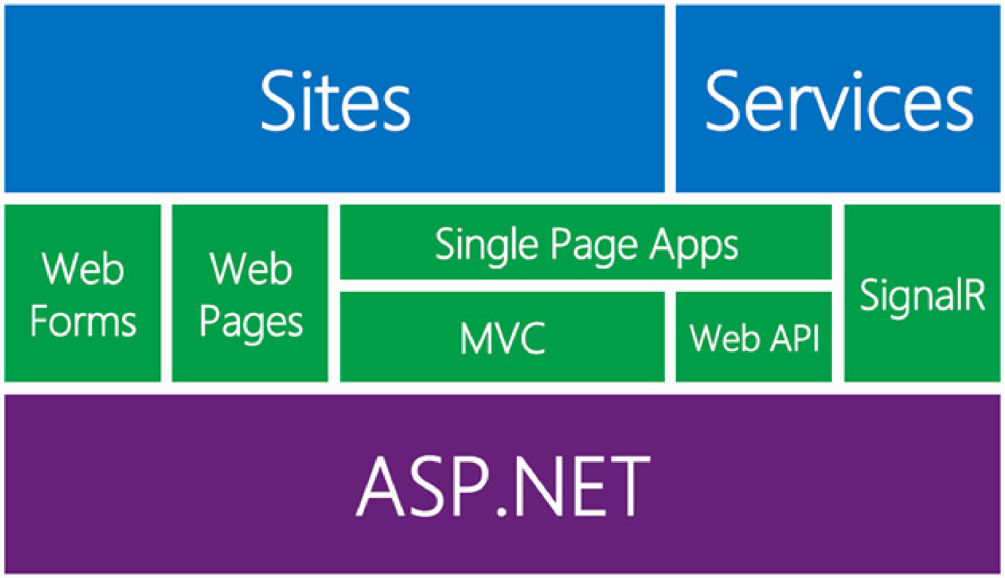
.NET?

- The Dot Net (.NET) is a open-source and cross-platform software development framework
  - integrates a number of technologies emerged from Microsoft to develop, run, and deploy web services and web or Windows applications.
- It is used to simplify the development process of multi-tier and client/server applications.
- Developers can develop big business or enterprise applications without writing code to manage security, pooling or transactions.



ASP.NET?

- The ASP.NET is a part of the .NET Framework which is used to create dynamic web pages.
- It is the latest evolution of server-side technologies and the successor to the classic Active Server Pages (ASP).
- It also simplifies the tasks of development, debugging, and deployment of web applications.



ASP.NET MVC – MVC stands for Model View Controller and allows to build web pages according to Model, View and Controller design pattern

ASP.NET Web Pages – Allows adding dynamic code and data access directly inside HTML

ASP.NET Web Forms – Allows building modular pages out of components with UI events

ASP.NET Web API – Allows developing web APIs on top of .NET framework

MICROSOFT .NET WEB STACK

- ASP.NET Web Forms was developed for two types of users:
  - **Developers who had experience** with classic Active Server Page (ASP) and were already building dynamic web sites mixing HTML and server-side code in Jscript. They were also used to interacting with the underlying HTTP connection and web server via abstractions provided by the core objects.
  - **Developers who were coming from the traditional WinForm** application development. They didn’t know anything about HTML or the web and were used to building applications by dragging UI components on a design surface.

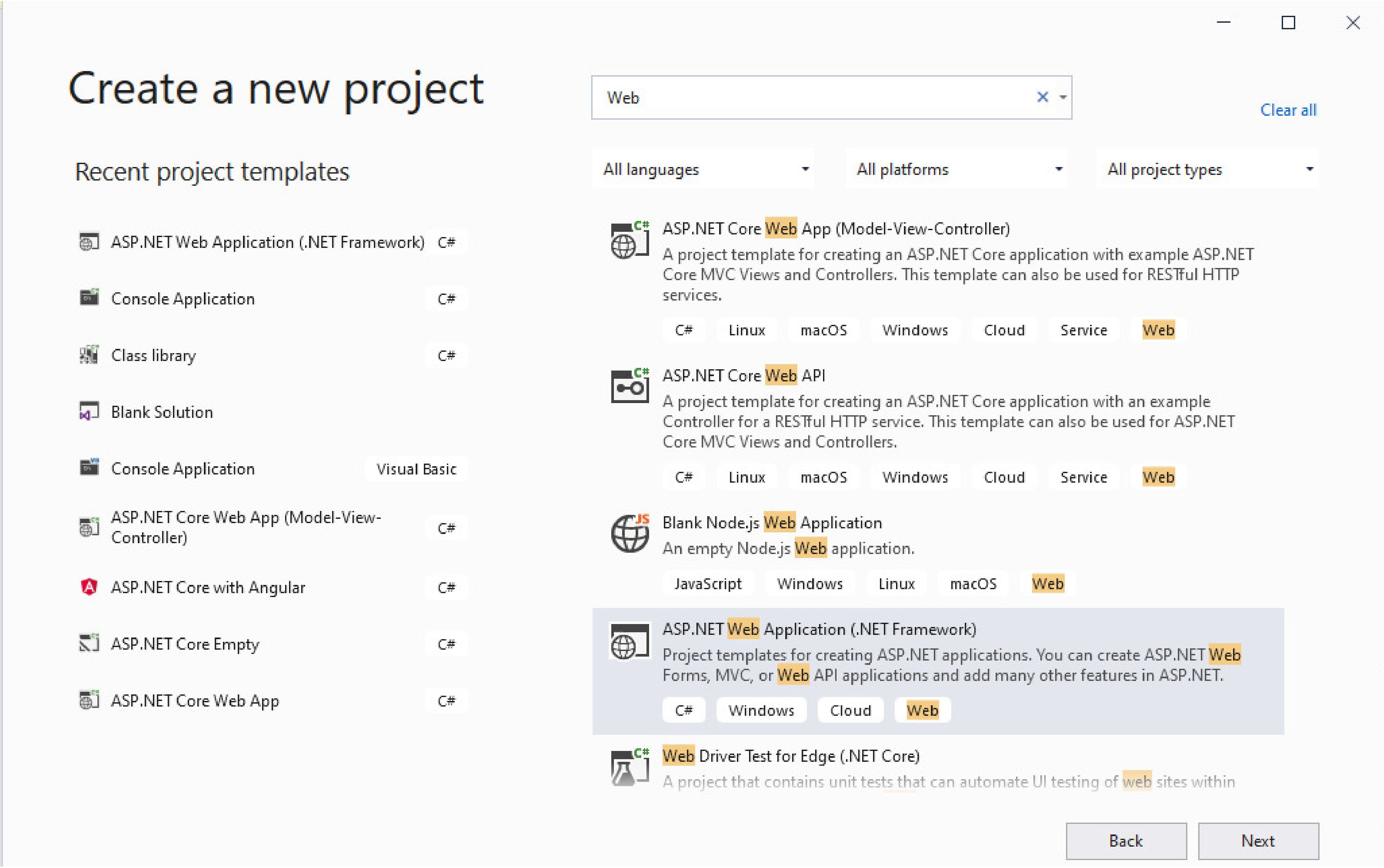
Limitations of Web Forms

All the core web abstractions were delivered within the **System.Web** library, and all the other web features depended on it.

- The .NET framework and Visual Studio were intimately tied.
  - For this reason, ASP.NET had to follow the release cycle of the other products, meaning that years passed between major releases.
- ASP.NET only worked with Microsoft's web server, Internet Information Services (IIS).
- Unit testing was almost impossible and only achievable using libraries that changed the way Web Forms worked.

- The problem of release cycles:
  - Other frameworks and languages pushing the web development evolution
  - Microsoft struggle to follow them. All are very small components
    - However, ASP.NET was huge and difficult to update
- The development style was needed to change:
  - Hiding and abstracting away the complexities of HTTP and HTML markup
    - helped a lot of WinForm developers to become web developers
  - Developer (after 5 yrs.) demands more control, specially the markup rendered on pages.

Example1: Hello World



Create a new project

Web

Recent project templates

- ASP.NET Web Application (.NET Framework) C#
- Console Application C#
- Class library C#
- Blank Solution
- Console Application Visual Basic
- ASP.NET Core Web App (Model-View-Controller) C#
- ASP.NET Core with Angular C#
- ASP.NET Core Empty C#
- ASP.NET Core Web App C#

ASP.NET Core Web App (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.
C# Linux macOS Windows Cloud Service Web

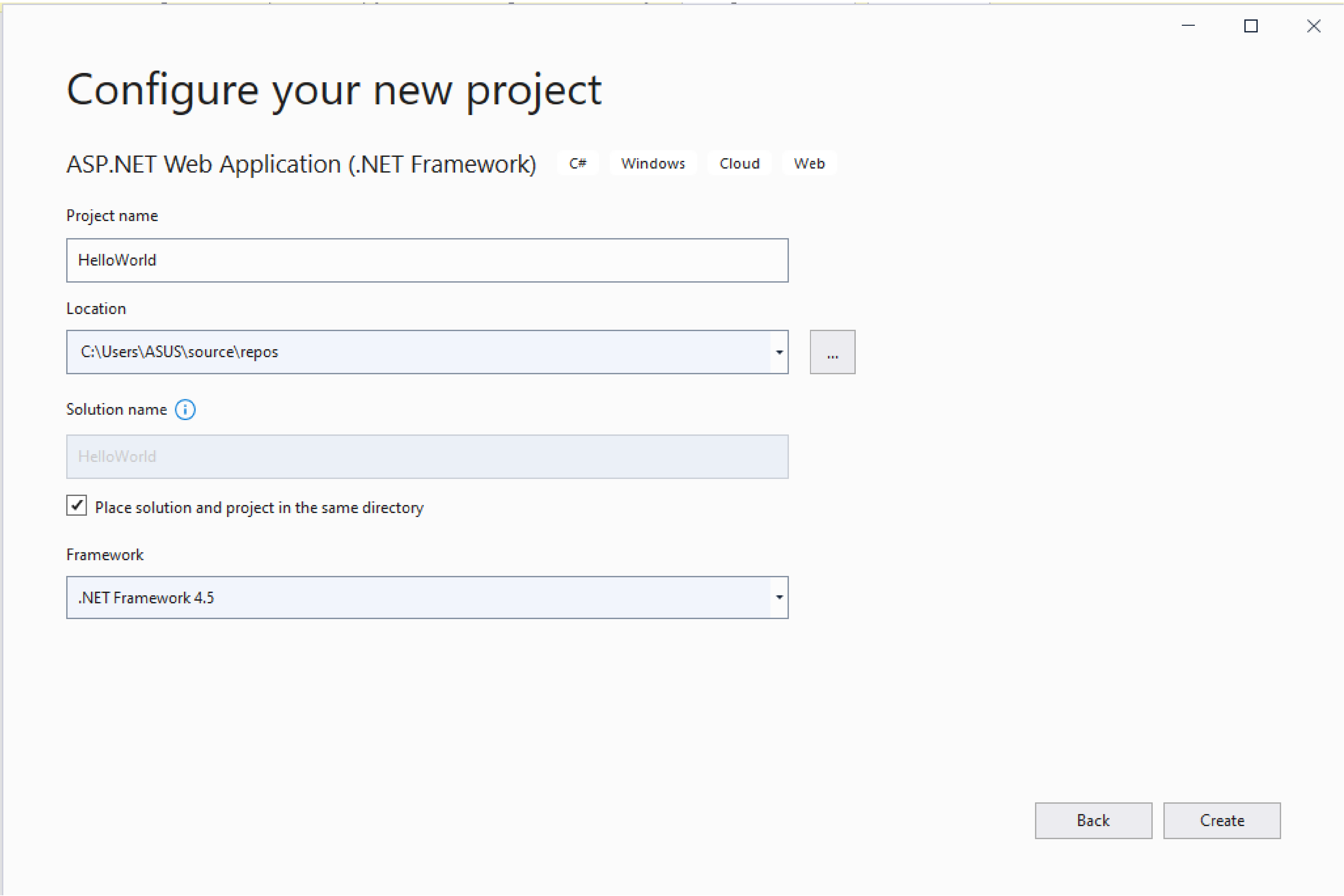
ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# Linux macOS Windows Cloud Service Web

Blank Node.js Web Application
An empty Node.js Web application.
JavaScript Windows Linux macOS 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

Example1: Hello World



Configure your new project

ASP.NET Web Application (.NET Framework)

Project name

HelloWorld

Location

C:\Users\ASUS\source\repos

Solution name

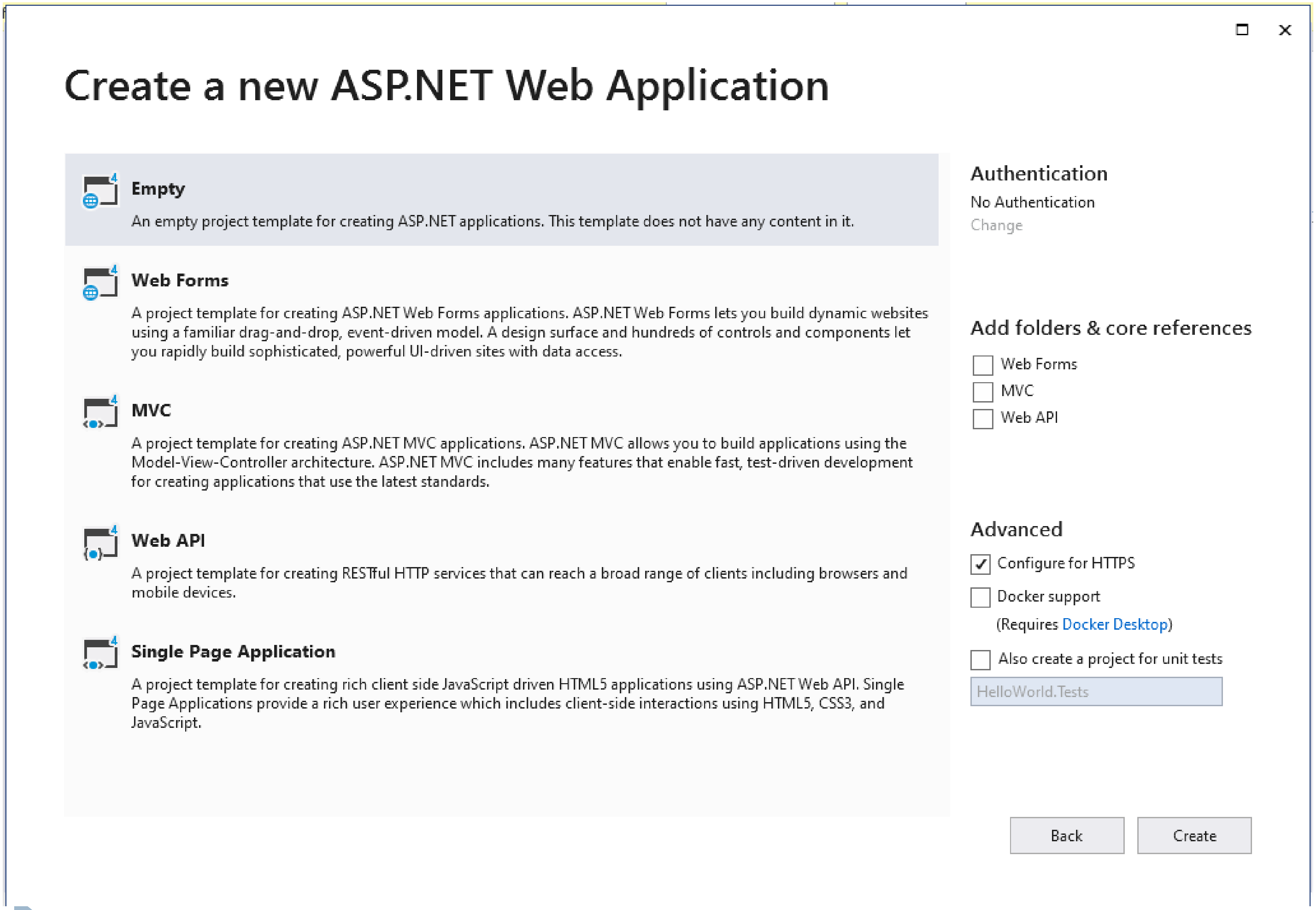
HelloWorld

☑ Place solution and project in the same directory

Framework

.NET Framework 4.5

Example1: Hello World



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.

Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application

A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication

No Authentication

Change

Add folders & core references

□ Web Forms

□ MVC

□ Web API

Advanced

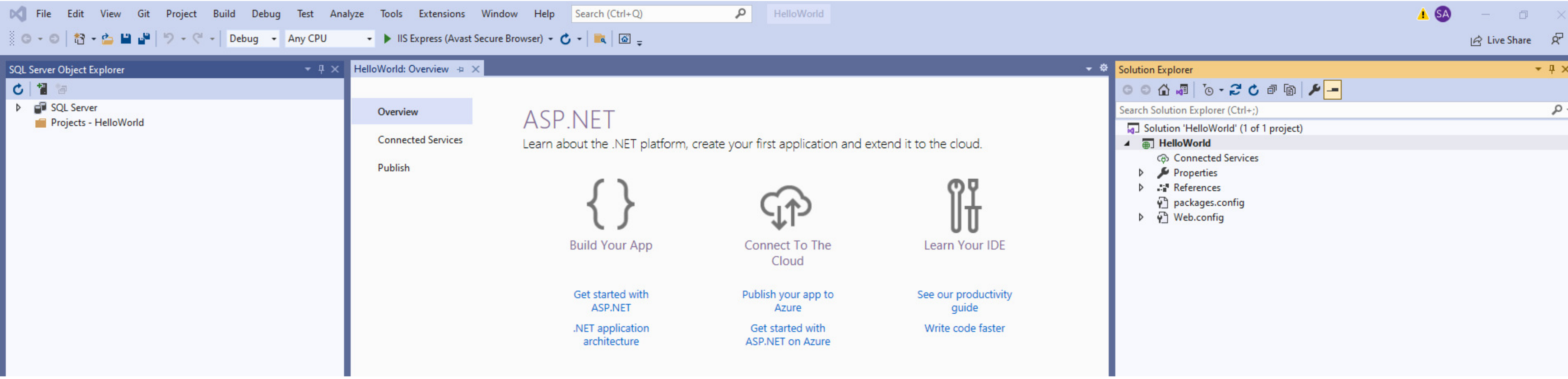
☑ Configure for HTTPS

□ Docker support

(Requires Docker Desktop)

□ Also create a project for unit tests

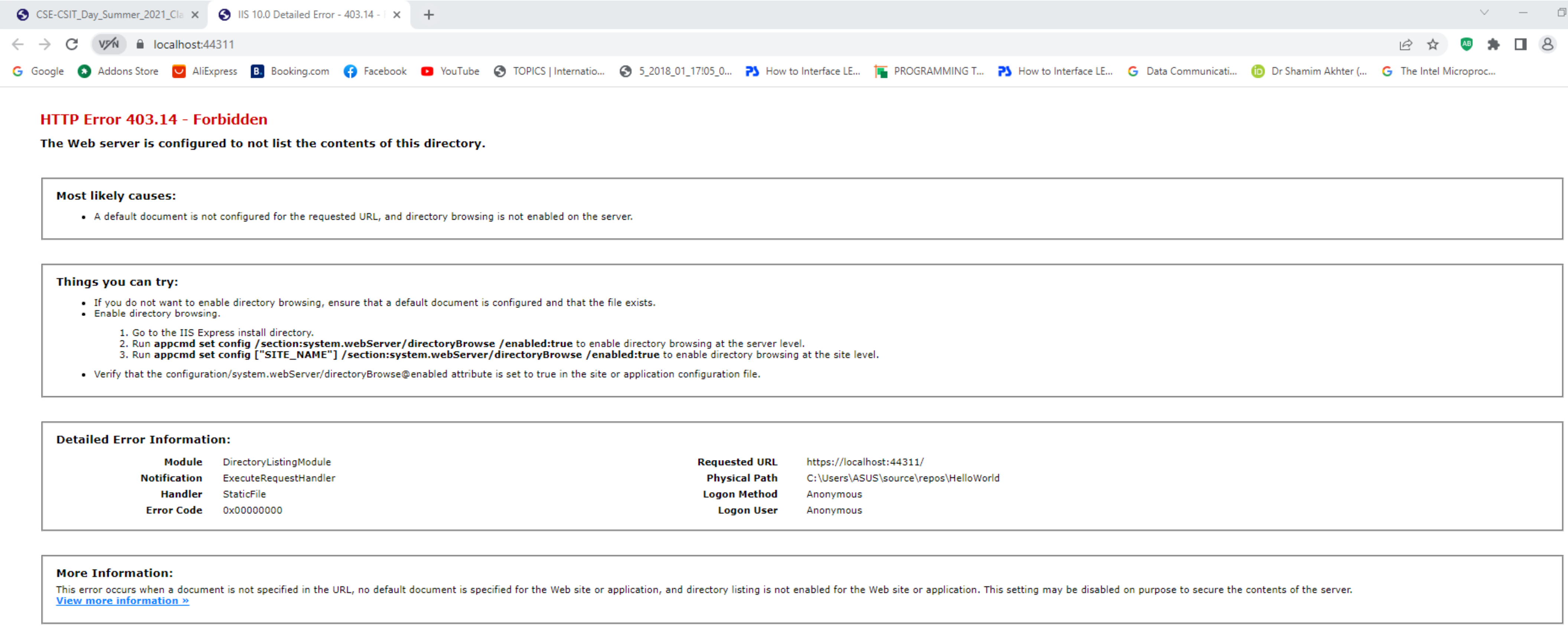
HelloWorld.Tests



ASP.NET

Learn about the .NET platform, create your first application and extend it to the cloud.

Build Your App
Connect To The Cloud
Learn Your IDE
Get started with ASP.NET
.NET application architecture
Publish your app to Azure
Get started with ASP.NET on Azure
See our productivity guide
Write code faster



HTTP Error 403.14 - Forbidden

The Web server is configured to not list the contents of this directory.

Most likely causes:

- A default document is not configured for the requested URL, and directory browsing is not enabled on the server.

Things you can try:

- If you do not want to enable directory browsing, ensure that a default document is configured and that the file exists.
- Enable directory browsing.
  1. Go to the IIS Express install directory.
  2. Run **appcmd set config /section:system.webServer/directoryBrowse /enabled:true** to enable directory browsing at the server level.
  3. Run **appcmd set config ["SITE_NAME"] /section:system.webServer/directoryBrowse /enabled:true** to enable directory browsing at the site level.
- Verify that the configuration/system.webServer/directoryBrowse@enabled attribute is set to true in the site or application configuration file.

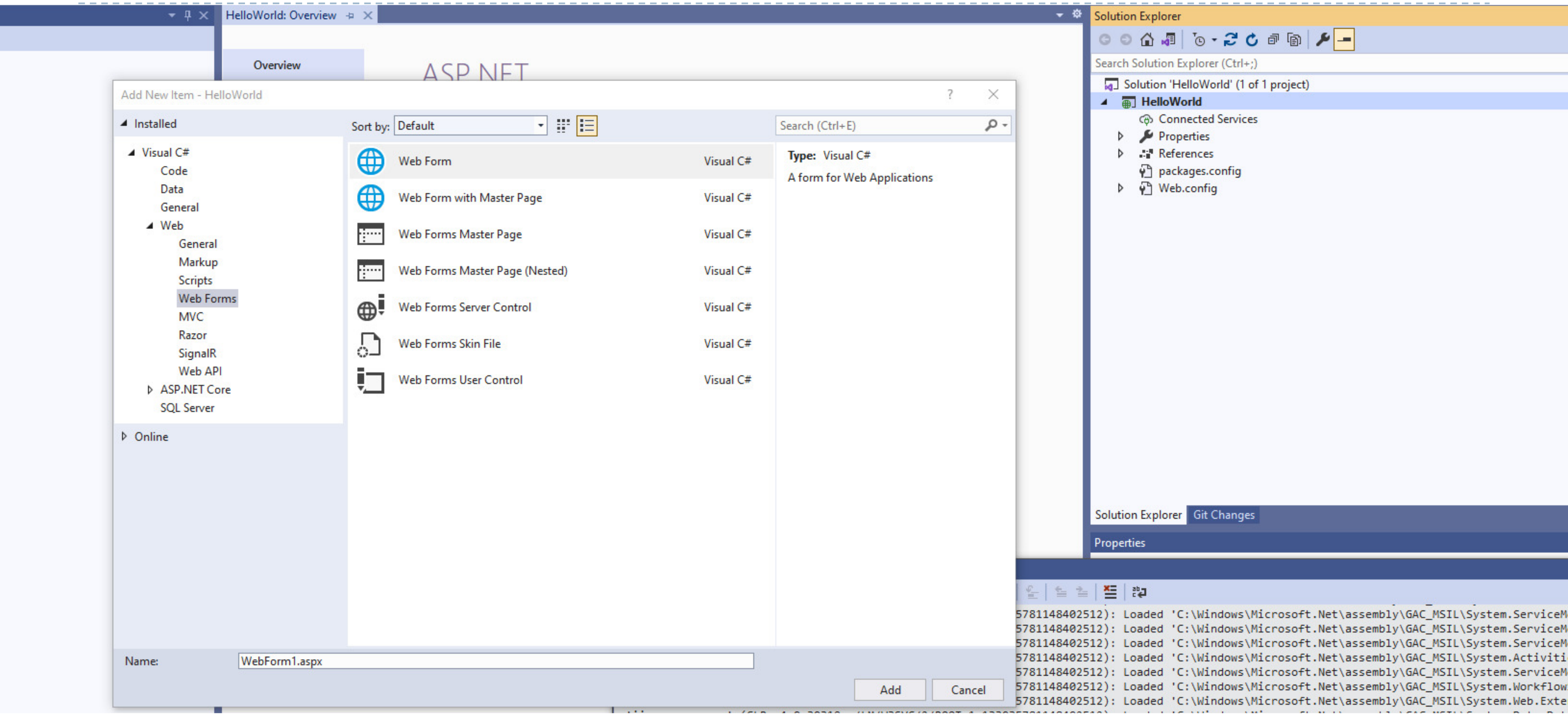
Detailed Error Information:

Module	DirectoryListingModule	**Requested URL**	https://localhost:44311/
Notification	ExecuteRequestHandler	**Physical Path**	C:\Users\ASUS\source\repos\HelloWorld
Handler	StaticFile	**Logon Method**	Anonymous
Error Code	0x00000000	**Logon User**	Anonymous

More Information:

This error occurs when a document is not specified in the URL, no default document is specified for the Web site or application, and directory listing is not enabled for the Web site or application. This setting may be disabled on purpose to secure the contents of the server.
View more information »

Add->New Item->Web Form



Type: Visual C#

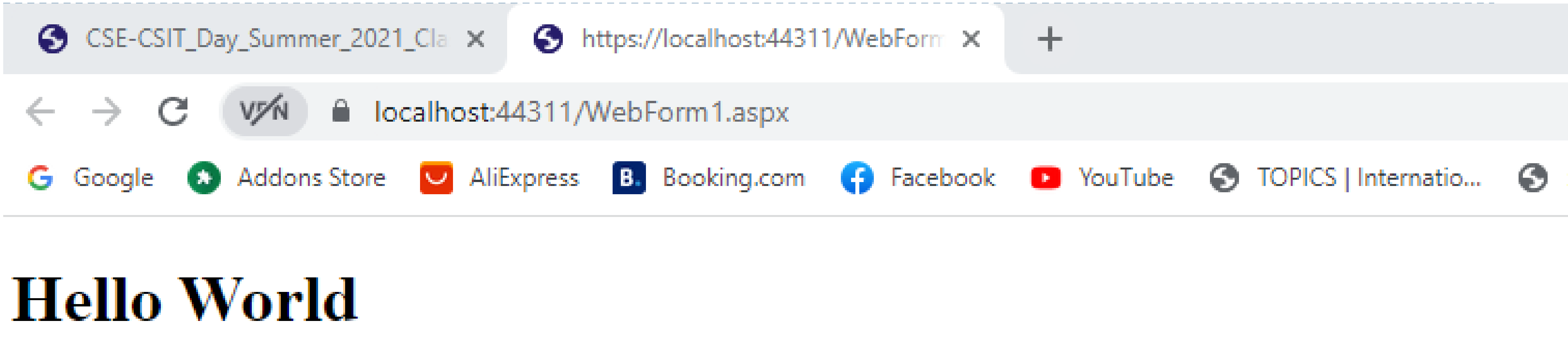
A form for Web Applications

Name: WebForm1.aspx

```
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.ServiceM
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.ServiceM
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.ServiceM
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Activiti
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.ServiceM
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Workflow
5781148402512): Loaded 'C:\Windows\Microsoft.Net\assembly\GAC_MSIL\System.Web.Exte
```

```
<body> <form id="form1" runat="server">
        <div>
            <h1> Hello World </h1>
        </div>
    </form>
</body>
```

Output



CSE-CSIT_Day_Summer_2021_Cla
https://localhost:44311/WebForm
localhost:44311/WebForm1.aspx
Google
Addons Store
AliExpress
Booking.com
Facebook
YouTube
TOPICS | Internatio...

Hello World

ASP.NET MVC

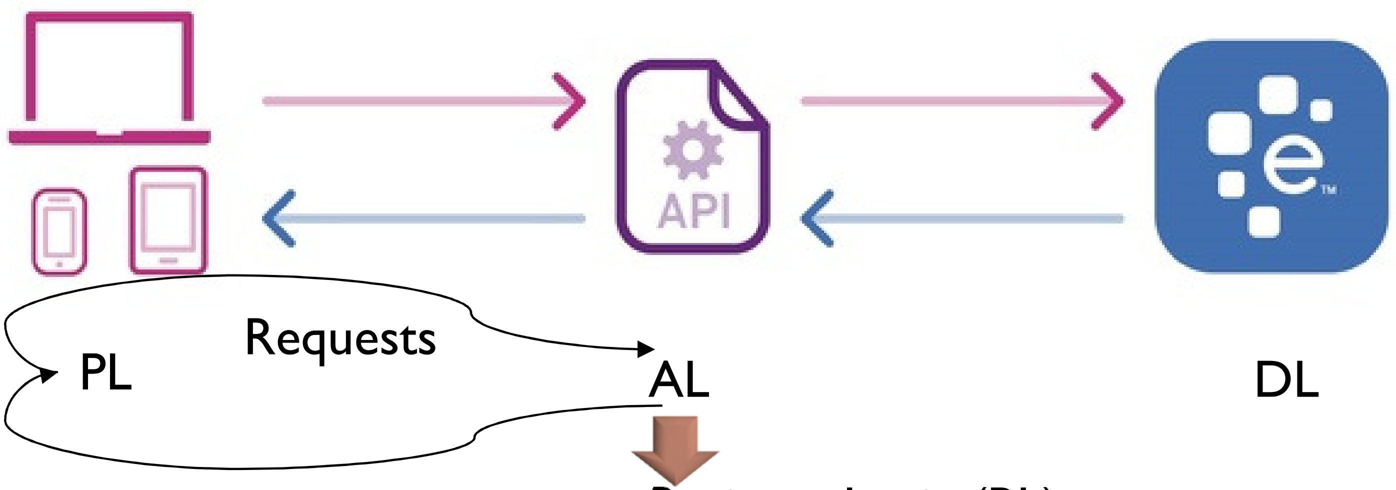
- Model-View-Controller design pattern
- A cleaner and better separation of business and presentation logic,
- Remove the server-side UI components, it gave complete control of the HTML markup to the developers.
- Furthermore, instead of being included inside the .NET framework, it was released out of band, making faster and more frequent releases possible.
- Limitation:
  - Although the ASP.NET MVC framework solved most of the problems of Web Forms, it still depended on IIS and the web abstracting library System.Web. This means that it was still not possible to have a web framework that was totally independent from the larger .NET framework.

ASP.NET Web API

- Build an even more modular component model,
  - finally ditching **System.Web** and creating a web framework that could live its own life independently from the rest of ASP.NET and the larger .NET framework.
  - A big role was also played by the introduction of NuGet, Microsoft's package distribution system, making it possible to deliver all these components to developers in a managed and sustainable way.
  - One additional advantage of the break-up from System.Web was the capability to not depend on IIS anymore and to run inside custom hosts and possibly other web servers.

Web API

- stands for **Application Programming Interface**



PL

Requests

AL

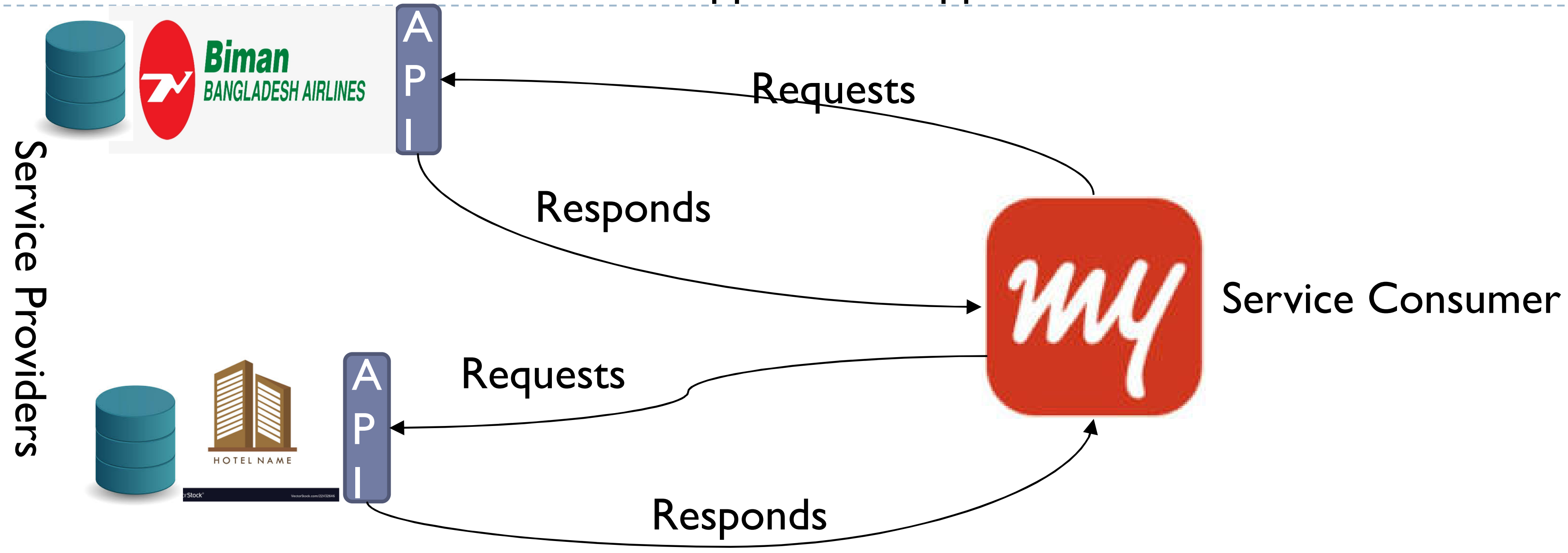
DL

Responds

Business Logic (BL)
Present them as functions or modules
-API, part of application layer and consist program
-Enable communication or exchange data between two software systems

How API Works

Communication
Application- Application



Service Providers

API

Requests

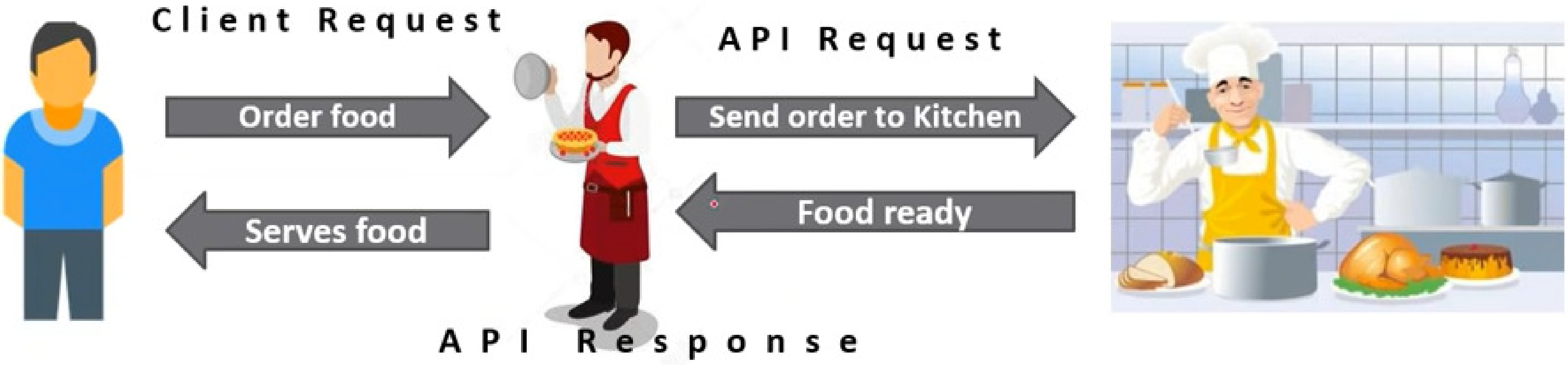
Responds

Service Consumer

API

Requests

Responds



Client Request

Order food

Serves food

API Request

Send order to Kitchen

Food ready

API Response

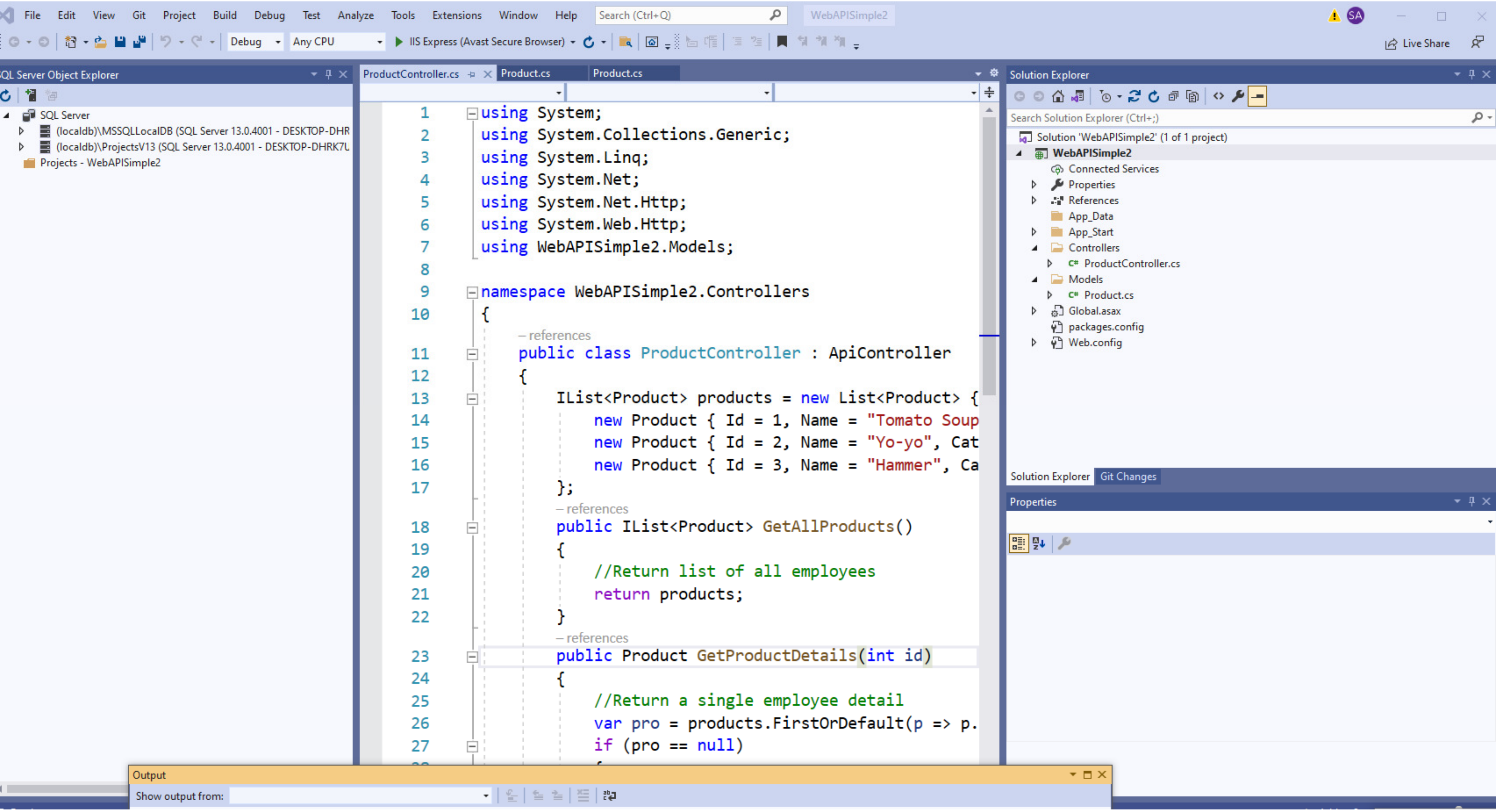
APIs are available in the Internet is called Web Services

Auto-implemented properties

- Auto-implemented properties in C# make code more readable and clean if there is no additional calculation needed. Compiler creates a private anonymous field that can only be accessed through the get and set accessors.

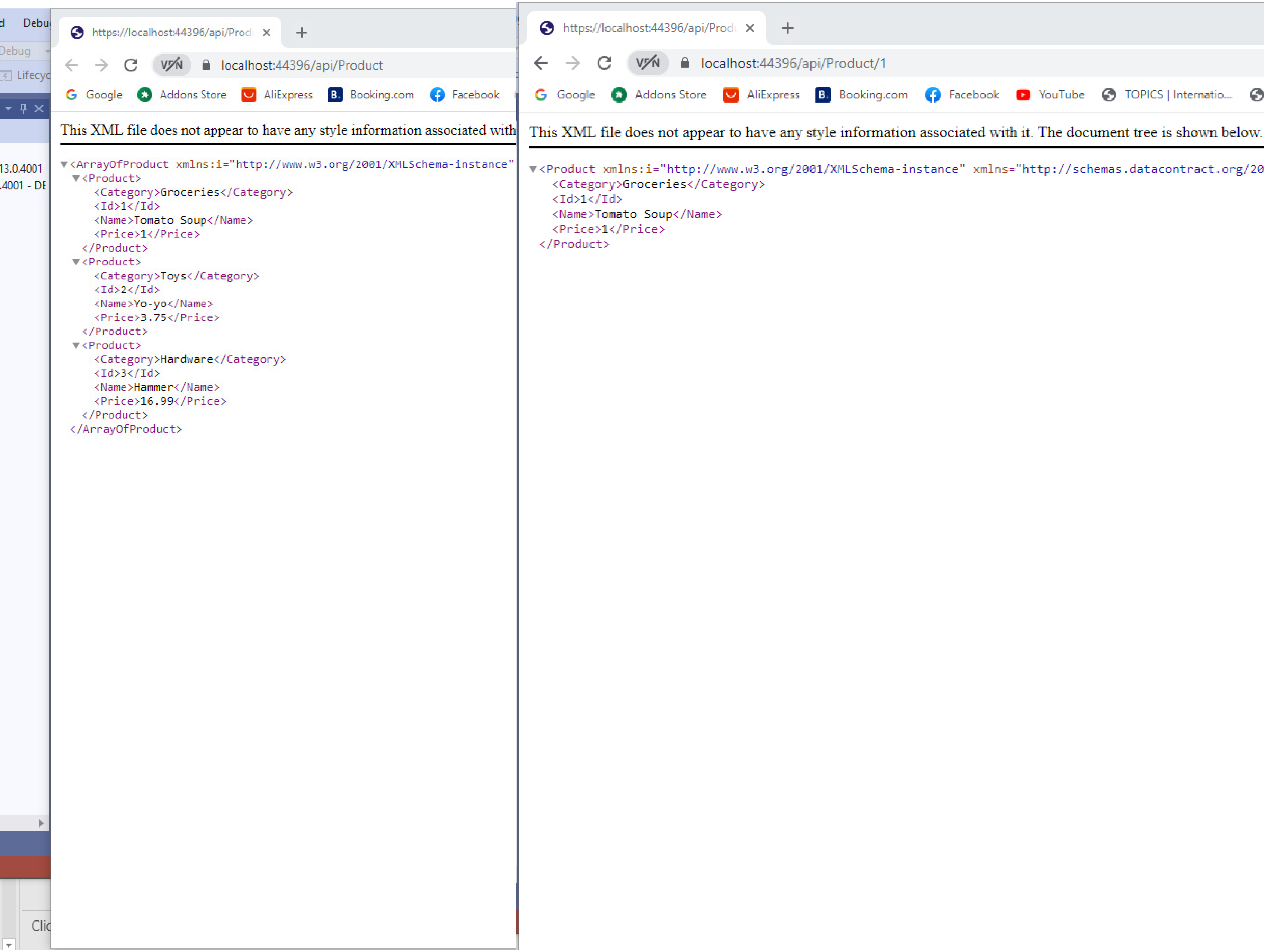
```
public int Id
       {
           get;
           set;
       }
public string Name
       {
           get;
           set;
       }
```

Example: Web API



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using WebAPISimple2.Models;

namespace WebAPISimple2.Controllers
{
    – references
    public class ProductController : ApiController
    {
        IList<Product> products = new List<Product> {
            new Product { Id = 1, Name = "Tomato Soup
            new Product { Id = 2, Name = "Yo-yo", Cat
            new Product { Id = 3, Name = "Hammer", Ca
        };
        – references
        public IList<Product> GetAllProducts()
        {
            //Return list of all employees
            return products;
        }
        – references
        public Product GetProductDetails(int id)
        {
            //Return a single employee detail
            var pro = products.FirstOrDefault(p => p.
            if (pro == null)
```


https://localhost:44396/api/Prod
localhost:44396/api/Product
Google
Addons Store
AliExpress
Booking.com
Facebook
https://localhost:44396/api/Prod
localhost:44396/api/Product/1
YouTube
TOPICS | Internatio...
Debug
Lifecyc
13.0.4001
.4001 - DE
Clic

This XML file does not appear to have any style information associated with

```
▼<ArrayOfProduct xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
  ▼<Product>
     <Category>Groceries</Category>
     <Id>1</Id>
     <Name>Tomato Soup</Name>
     <Price>1</Price>
   </Product>
  ▼<Product>
     <Category>Toys</Category>
     <Id>2</Id>
     <Name>Yo-yo</Name>
     <Price>3.75</Price>
   </Product>
  ▼<Product>
     <Category>Hardware</Category>
     <Id>3</Id>
     <Name>Hammer</Name>
     <Price>16.99</Price>
   </Product>
 </ArrayOfProduct>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Product xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/20
   <Category>Groceries</Category>
   <Id>1</Id>
   <Name>Tomato Soup</Name>
   <Price>1</Price>
 </Product>
```

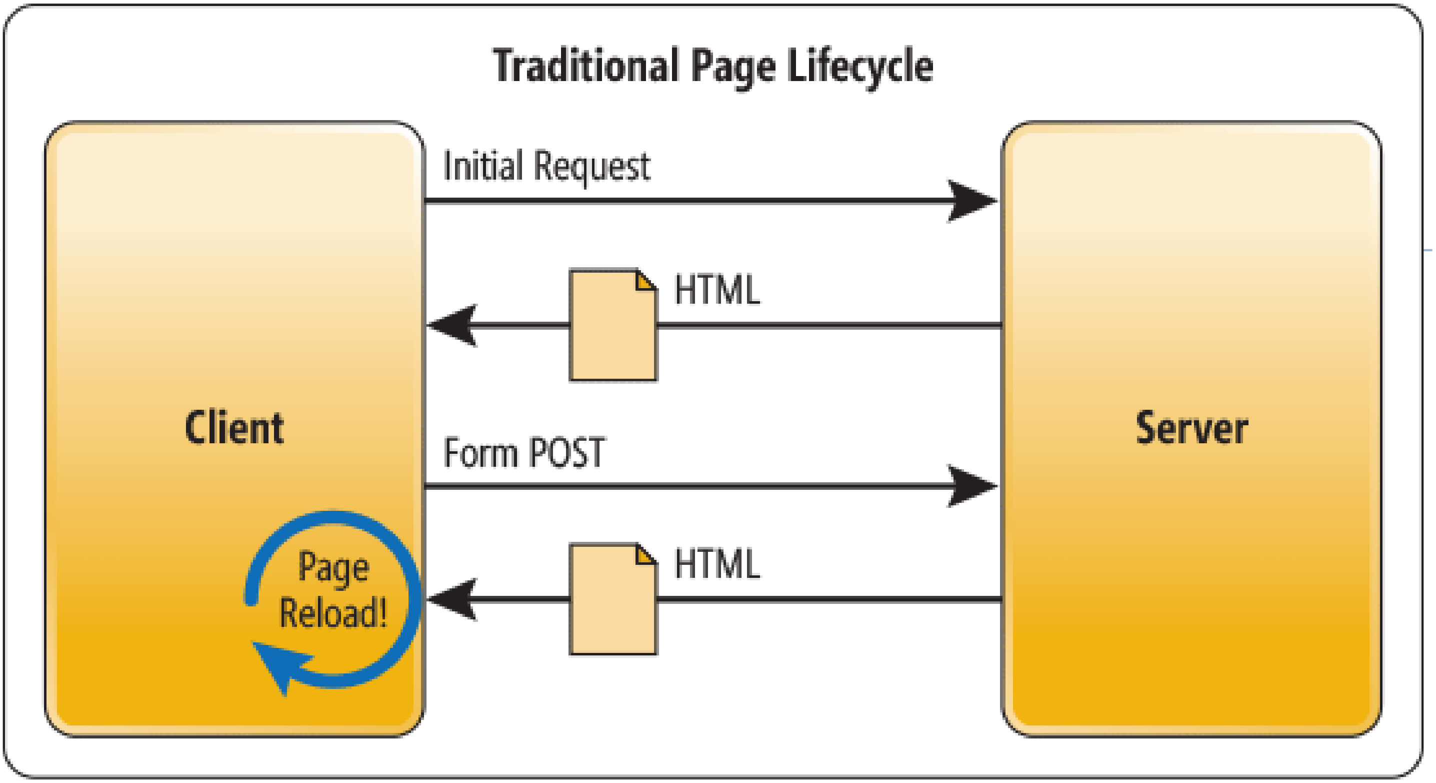
Example

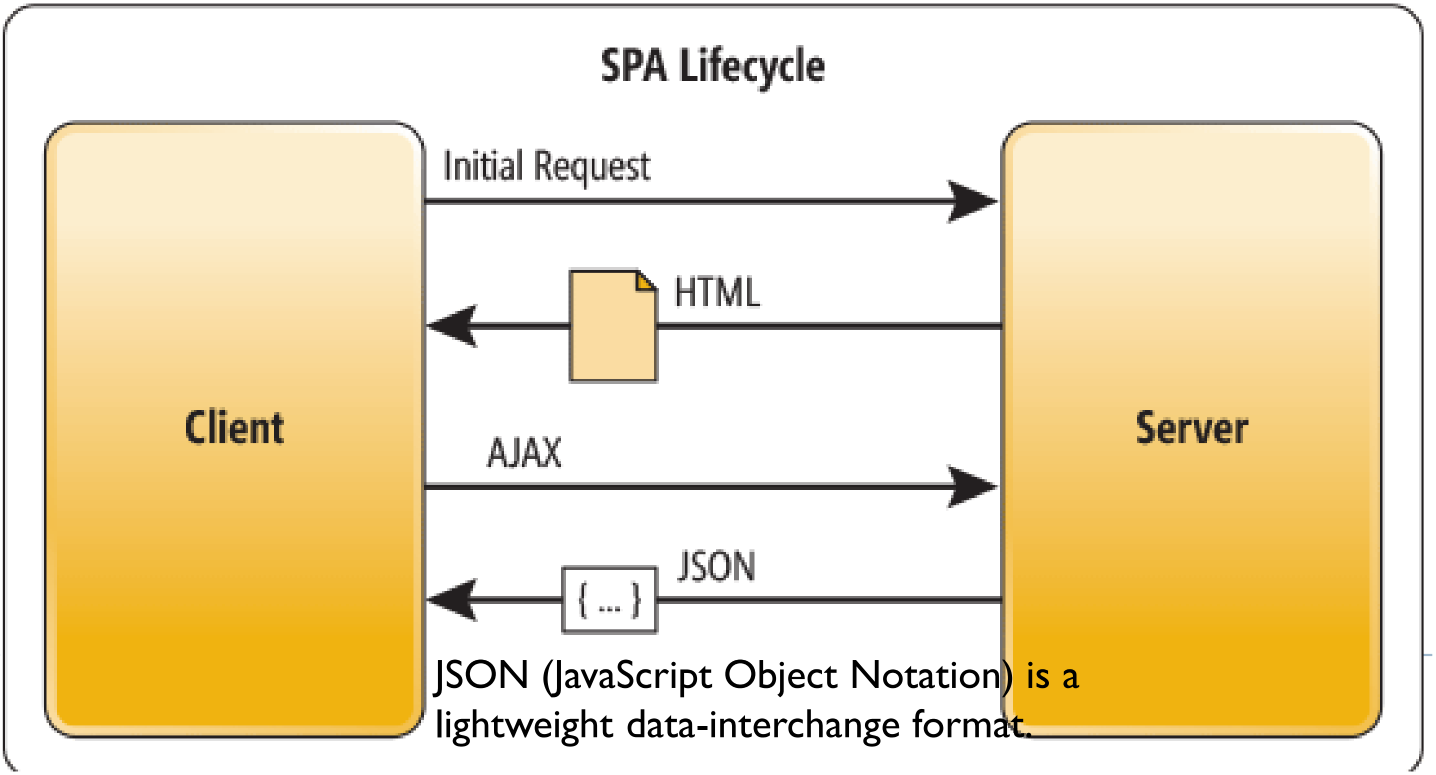
- https://www.c-sharpcorner.com/UploadFile/8a67c0/getting-started-with-web-api-step-by-step-with-sample-applic/

Single Page Applications (SPAs).

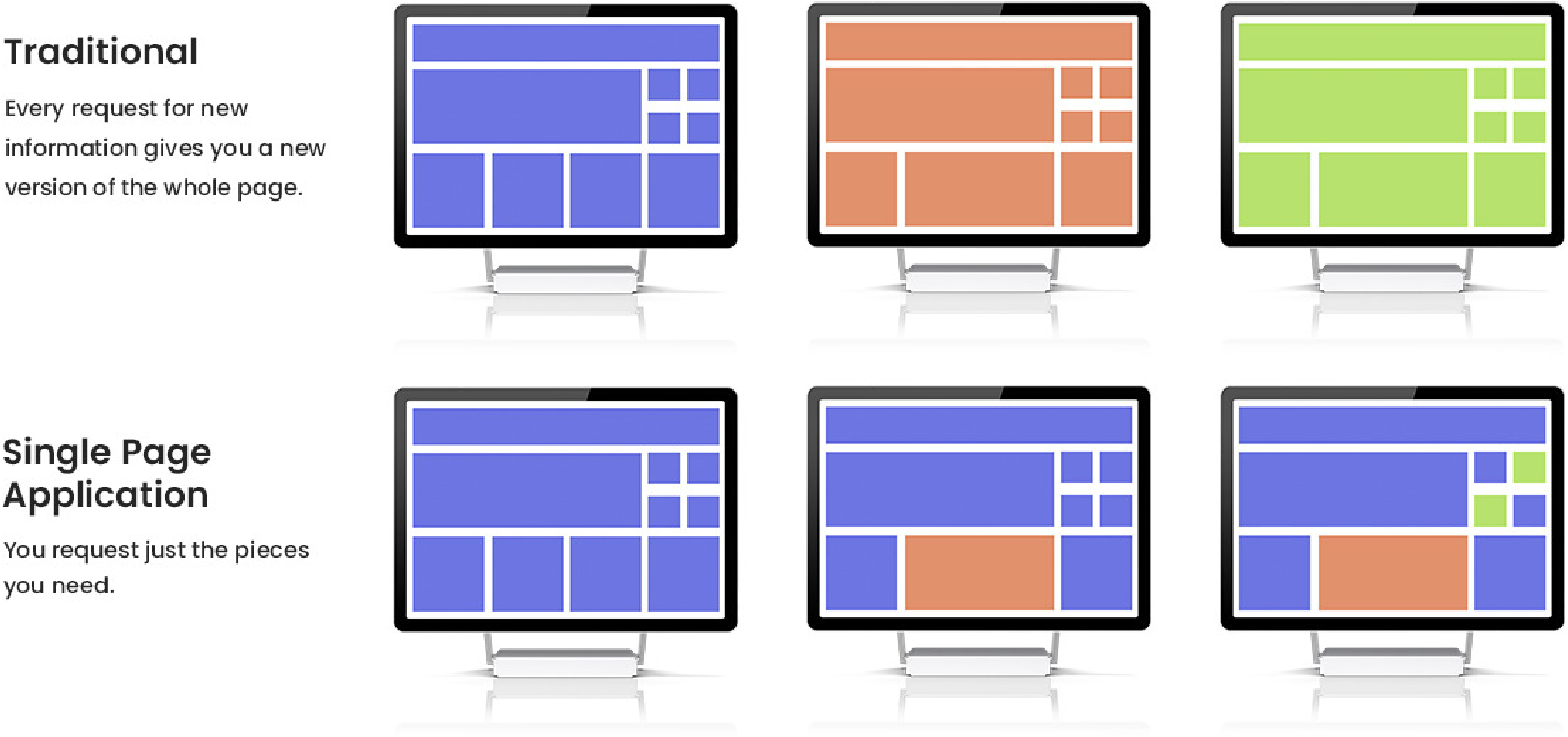
- Instead of interconnected, server-generated, data-driven pages, applications were becoming mostly static pages where data was displayed interacting with the server via AJAX(Asynchronous JavaScript and XML) calls to web services or Web APIs. Also, many services started releasing APIs for mobile apps or third-party apps to interact with their data.

As the names suggests, it is a single page where all information is available on the same page.





JSON (JavaScript Object Notation) is a lightweight data-interchange format.



Traditional

Every request for new information gives you a new version of the whole page.

Single Page Application

You request just the pieces you need.

A single-page application (SPA) is an app that works inside the browser and reduces reloading requirements. It is one page that is visited and then with the aid of JavaScript, it is loaded all other content.

- These types of applications are accessible and avoid the waiting time for the user. Examples are Gmail, Google Maps and Facebook.

OWIN and Katana @.Net Framework

- System.Web is something that has existed ever since ASP (non .NET version) and internally contains many things that you might not even need (such as Web Forms or URL Authorization), which by default all run on every request, thus consuming resources and making ASP.NET applications in general lot slower than its counterparts such as Node.js for example.
- Open Web Interface for .NET (OWIN) is a specification on how web servers and web applications should be built in order to decouple them and allow movement of ASP.NET applications to environments which were not supported before.
- Katana on the other hand, is a fully developed framework made to make a bridge between current ASP.NET frameworks and OWIN specification. At the moment, Katana has successfully adapted the following ASP.NET frameworks to OWIN:
  - Web API
  - Signal R
- ASP.NET MVC and Web Forms are still running exclusively via System.Web, and in the long run there is a plan to decouple those as well.

- Katana is slowly getting retired. Version 3.0 will most likely be last major release of Katana as a standalone framework.

- ASP.NET 5 is the successor to Katana. Katana was the beginning of the break away from System.Web and to more modular components for the web stack. You can see vNext/5 as a continuation of that work but going much further (new CLR, new Project System, new http abstractions). ASP.NET 5 was built on top of .NET Core 5. NET Core 5 was lightweight factored version of .NET Framework.

- After ASP.NET 5 ASP.NET Core 1.0 and .NET Core 1.0 were introduced.