

Module – 5

Introduction to Web Dev

- ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.
- ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.
- ASP.NET is a part of Microsoft .Net platform.
- ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework.
- These codes can use the entire hierarchy of classes in .Net framework.
- ASP.NET is built on the CLR (Common Language Runtime) which allows the programmers to execute its code using any .NET language (C#, VB etc.).
- It is specially designed to work with HTTP and for web developers to create dynamic web pages, web applications, web sites, and web services as it provides a good integration of HTML, CSS, and JavaScript.
- .NET Framework is used to create a variety of applications and services like Console, Web, and Windows, etc.
- ASP.NET is only used to create web applications and web services.
- That's why we termed ASP.NET as a subset of the .NET Framework.

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

ASP.NET Web Forms Model

- ASP.NET web forms extend the event-driven model of interaction to the web applications.
- The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.
- All clientside user activities are forwarded to the server for stateful processing.
- The server processes the output of the client actions and triggers the reactions.
- Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:
 - Page state
 - Session state

ASP.NET Web Forms Features

- Server Controls
- Master Pages
- Working with data
- Membership
- Client Script and Client Frameworks
- Routing
- State Management
- Security

ASP.NET Component Model

- The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:
 - Serverside counterparts of almost all HTML elements or tags, such as <form> and <input>.
 - Server controls, which help in developing complex user-interface. For example, the Calendar control or the Grid view control.
- ASP.NET is a technology, which works on the .Net framework that contains all web-related functionalities.
- The .Net framework is made of an object-oriented hierarchy.
- An ASP.NET web application is made of pages.
- When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system.
- The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework.
- Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

ASP.NET MVC

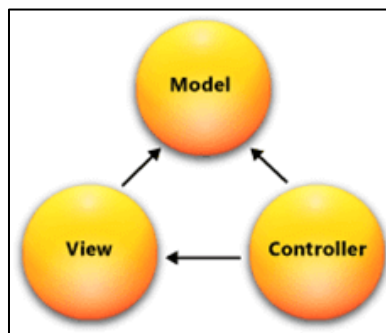
The MVC (Model-View-Controller) is an application development pattern or design pattern which separates an application into three main components:

- Model
- View
- Controller

Model: Model is a part of the application which implements the logic for the data domain of the application. It is used to retrieve and store model state in a database such as SQL Server database. It also used for business logic separation from the data in the application.

View: View is a component that forms the application's user interface. It is used to create web pages for the application. An example would be an edit view of a Products table that displays text boxes, drop-down lists and check boxes based on the current state of a Product object.

Controller: Controller is the component which handles user interaction. It works with the model and selects the view to render the web page. In an MVC application, the view only displays information whereas the controller handles and responds to the user input and requests.



Advantages of ASP.NET MVC Framework

This approach provides the following advantages.

- It manages application complexity by dividing an application into the model, view and controller.
- It does not use view state or server-based forms. This makes the MVC framework ideal for developers who want full control over the behavior of an application.
- It provides better support for test-driven development.
- It is suitable for large scale developer team and web applications.
- It provides high degree of control to the developer over the application behavior.

Rest Web API

- Web API is the enhanced form of the web application to provide services on different devices like laptop, mobile, and others.
- Today, all kind of businesses use the internet as a cost-effective way to expand their business in the international market.
- Web application helps to exchange information on the internet and also helps to perform a secure transaction on web sites.

- Web applications are popular as the web browser is available in default, we don't need any installation of software on computers with operating systems.
- For example, Facebook, Flickr, and Wikipedia are majorly used example of a web application.

Technically, a web application consists of two types of scripts:

1) **Client-side scripts:** JavaScript, HTML, and other client-side scripting languages are used to design the web forms to present information to users.

2) **Server-side scripts:** ASP and other server-side scripting languages are used to perform business logic and database related operations like storing and retrieving information.

RESTful services

- Web API is the enhanced form of a web application.
- SOAP (Simple Object Access Protocol) was an XML based protocol for developing the connected web applications.
- Problem with the SOAP was that with each request, Metadata is attached with data to be transferred.
- This Metadata converts small data to heavy data on the server.
- Web API may or may not be RESTful services, but they are always HTTP based services.
- REST stands for Representational State Transfer.
- In REST API, only the state of the object is sent to the server to find the desired result.
- REST is an architectural pattern for developing an API that uses HTTP as its underlying communication method.

Principles of REST API

The six principles of REST API are:

- 1) Stateless
 - 2) Client-Server
 - 3) Uniform Interface
 - 4) Cacheable
 - 5) Layered System
 - 6) Code on demand
- 1) **Stateless:** When the request from the client is sent to the server, it contains all the required information to make the server process it. A request may be part of QueryString or URL.

2) Client-Server: Separating the functionality helps to increase user interface portability across multiple platforms as well as extended the scalability of the server components.

3) Uniform Interface: To obtain the uniformity throughout the application, REST has defined four interface constraints for which are:

- Resource Identification
- Resource Manipulation using representations
- Self-descriptive messages
- And hypermedia as the engine of the web application

4) Cacheable: In order to provide a better performance, applications are made cacheable. It is done by marking the response as cacheable or non-cacheable implicitly or explicitly. If the resource is defined cacheable, then the client cache can reuse response data for equivalence requests.

5) Layered System: The layered system allows an application to be most stable by limiting component behavior. The layered architecture also helps to enhance security as the component at each layer cannot interact beyond each layer they are in.

6) Code on demand: Constraint that is used optionally and least. Analyze and simplify the client by creating a smart application which doesn't lie on its own code structure.

Methods of REST API

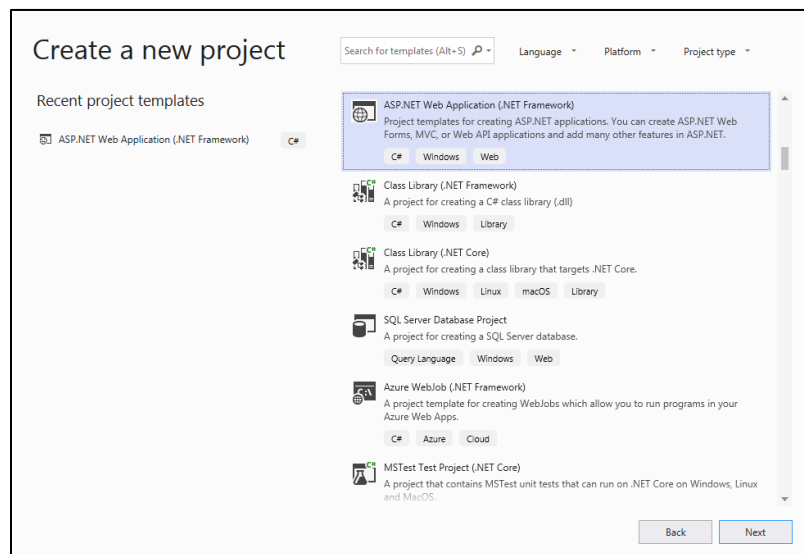
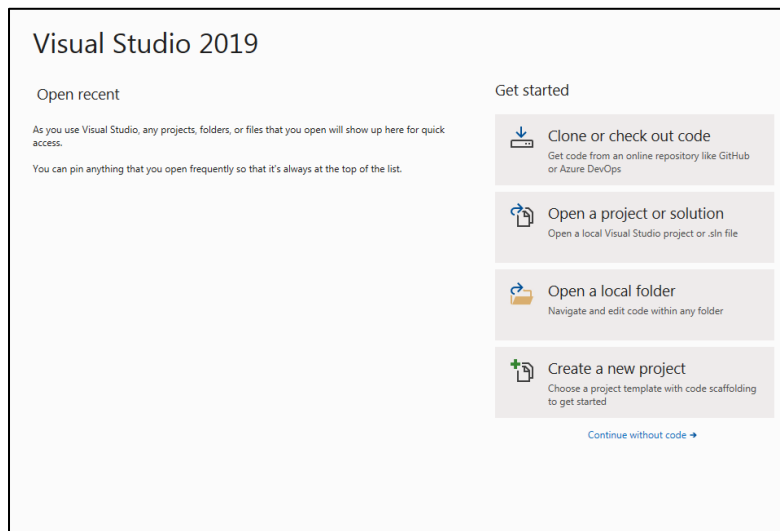
- Working on web technologies, we work on CRUD applications.
- In these applications, CRUD is to CREATE, READ, UPDATE, and DELETE a resource.
- To perform these actions like to create a resource, read a resource, update a resource, or delete a resource, we can use HTTP methods also called as REST methods.

26.Start With Project

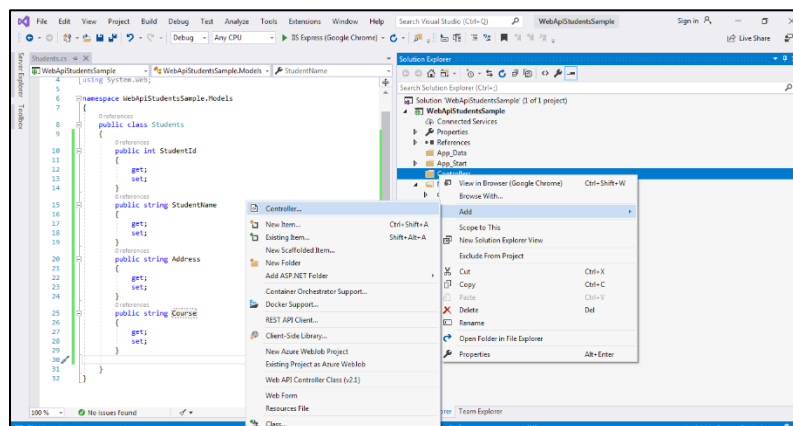
Create New Web API Project

Firstly open Visual Studio (here we have Visual Studio 2019).

Now select Create a new project:



- Now select the template ASP.NET Web Application (.NET Framework) with C# and name the project as "WebApiStudentsSample".
- In the ASP.NET Project dialog, select the Empty template and also check Web API option. Click OK.
- Add a class with the name "Student" to define properties and other business logic. We can also define other logics as validation, data access, etc.
- Right-click on Models, Select Add option, and then select Class and give a name to the class as "Student.cs".
- Controller class handles HTTP request from the client, which may be a desktop application, mobile device, and browser.
- Right-click on Controllers, Select Add option, and then select 'Controller'.



- To run a Web API, firstly press F5 or Ctrl+F5 or Click on IIS express run icon then the browser will open with the URL like <https://localhost:44329/>
- Now to find the list of all students edit the URL as <https://localhost:44329/api/student>

Create Controller And Model

- Creating Controller
- We right click on controller folder and click on Add then Add Controller.
- Then we name the controller and it is created.
- A controller can be created in both MVC and Web API
- In MVC it is executed in accordance to following URI.
- URI : {Controller}/{Action}/{Id}

- In Web API URI: API/{Controller}/{Id}
- Controller gets requests from the client and then it shows them the data according to the action performed.

Creating Model

- Models are the classes in the c# with fields and constraints mapped into the database as datatables.
- We right click on model folder and click on Add then Add class.
- Then we name the class and it is created.
- We can write an ApplicationDbContext file with references of all our model classes and then we configure it to the database to connect it to the database.
- Then we create object of that class in the controller to use it's properties

Parameter Passing

- When Web API calls a method on the controller it must set parameters by a process called parameter binding.
- So for simple datatypes like int, bool, double, decimal, string Web API tries to get values from URI by default.
- For complex datatypes like user defined objects it takes Web API tries to read value from message body.
- These are default cases but these methods can be overridden by for getting complex data from URI itself and for getting simple data from the message body.
- These are used just to override in given conditions otherwise in default case other methods can be used.

Serialization

- It is a process of storing objects in physical storage or send over a connection as a stream of bytes which after getting received by the recipient can be then be deserialized into an object again or when we want to read that object from the physical storage.
- The reverse process is deserialization.
- It is an technology that enables an object to be converted into a stream of data so they can be easily passes across the system or machine.
- This format should be understandable by both end of a communication channel.
- It is used by the web services , remoting for transmitting data between server and a client.
- The namespace of serialization contain formatter interface which contain the methods serialize and deserialize that can used to save and load data to and from a stream.
- In order to implement serilization in .NET ,we basically require a stream and a formatter.

- Stream act as a container for serializes object.
- Formatter is used to serialize these objects onto the stream.

Json Serialization

- As the name suggest it is used to convert object in to the JSON stream format.
- We can convert object in JSON stream format and also can get back that object from JSON stream using the concept of the serialization and de-serialization.
- The quickest method of converting between JSON text and a .NET object is using the JsonSerializer.
- The JsonSerializer converts .NET object into their JSON equivalent and back again by mapping the .NET object property names from the JSON property names and copies the values for you.

Routing

Routing is a process of mapping the browser request to the required controller's action method. Each MVC Application has default route for the HomeController. We can set custom routing for other newly made Controllers.

- Working:
 - When an URL's Request matches any registered route patterns the routing engine exacts it to the corresponding handler.
 - If it doesn't match then engine will generate corresponding error on the web page.
- URL for API: api/{controller}/{id}
- URL for MVC: {controller}/{action}/{id}

Types Of Routing

- Conventional Routing
 - Conventional or Traditional Routing also is a pattern matching system for URL that maps incoming request to the particular controller and action method.
 - We set all the routes in the RouteConfig file.
 - RouteConfig file is available in the App_Start folder.
 - We need to register all the routes to make them operational.
 - By giving this sort of routing every URL will match the default pattern given in the RouteConfig file to show the view if they don't then 404 error page is shown.
 - Advantage of it over attribute routing is that all the routing pattern is at one place which is generalized and mandatory for all paths to follow it makes routing simple.

- Attribute Routing:
 - As it uses attribute to write the routing so it's called attribute routing.
 - It can be applied to any Controller and Action method.
 - It helps customize an URL pattern according to the needs and that holds a major advantage over conventional routing.
 - As Conventional routing pattern is all at one place and rules need to be followed by all controllers and actions so there is no room if we require a different URL pattern for an action method. But with attribute routing we may generalize the portion which is common for a controller over a controller and then give routes to action methods according to the requirement.

Config

- A configuration file (web.config) is used to manage various settings that define a website. The settings are stored in XML files that are separate from your application code.
- In this way you can configure settings independently from your code.
- Generally, a website contains a single Web.config file stored inside the application root directory.

The behavior of an ASP.NET application is affected by different settings in the configuration files:

- machine.config
 - web.config
- The machine.config file contains default and the machine-specific value for all supported settings.
- The machine settings are controlled by the system administrator and applications are generally not given access to this file.
- An application however, can override the default values by creating web.config files in its roots folder.
- The web.config file is a subset of the machine.config file.
- If the application contains child directories, it can define a web.config file for each folder. Scope of each configuration file is determined in a hierarchical top-down manner.