

Name: Priya Gosai

Module-5

Topics:

1. Introduction to WEB development.

- I. Asp.net web forms**
- II. MVC**
- III. Rest web API**

2.Start with project:

- IV. Create new web API project**
 - V. Setting up infrastructure**
 - VI. Create controller, Model**
 - VII. Parameters(from URI ,from body)**
 - VIII. Serialization**
 - IX. Routing**
 - X. Config**
-

1. Introduction to WEB development.

I . Asp.net web forms

➤ Introduction to the Asp.net

- Asp.net is a open source and cross platform for web development.
- It provides services to allow the creation , deployment, and execution of web applications and web services.
- It has built-in web forms controls ,which can be used in creating web forms. We can also create our own user controls also.
- ASP.NET extends the .NET developer platform with tools and libraries specially for building web apps.
- It is a server side scripting language based on the IIS(internet information services)
- It is a browser independent.
- With the use of Asp.net framework we can build the application like web application, desktop application froms , web pages.
- The full form of ASP is : Active Server Pages

ASP.net provides three development styles for creating web application.

It means we can creating web application using three different styles or architectures.

1. Web forms
2. ASP.net MVC
3. ASP.net web pages

Web forms using Asp.net:

- We can develop web application using web forms.
 - Web form is an event driven development framework. There are many controls provided in the framework which are used for generating events like buttons ,text boxes ,labels etc. we can handle that event by writing event handler methods.
 - There is a large library of controls available in the ASP.NET framework which we can use to create web forms.some controls are similar to HTML like buttons, text boxes ,tables
 - It is used to develop application with powerful data access .
 - Web forms provides many options for storing ,retrieving and displaying data.
 - It provides server side controls and event to create web applications.
-

- It is a part of the ASP.NET framework.
- It can be written in any programming language which supports the common language runtime, such as c# or VB.NET
- It allows you to build dynamic website using drag and drop

ASP.NET MVC:

- MVC stands for Model View Controller.
- It is a design pattern for achieving a clean separation of concerns.
- Model – all the database related classes are in Model
- View – the front end of the web Application and its all functions are written in the view part
- Controller – it is the bridge between database and user interface. All the logic for how our web application will work are written in this part.

ASP.NET web pages

- It is used to create dynamic web pages.
- It provides fast and lightweight way to combine server code with html .
- It helps to add video, various links .
- It also provides other features like you can create beautiful sites that conform to the latest web standards.
- It uses razor syntax to insert dynamic content in the web pages

All these are standard and well equipped frameworks .we can create web application with any of them .there are also based on the .net framework and share core functionality of .net and ASP.net.

2. MVC

- it is an architectural pattern that separates an application into three main logical parts:
 1. the model
 2. the view
 3. the controller
- Each of these component are built to handle specific development are of an application.
- It is one of the most used industry standard web development framework to create scalable and extensible projects.
- As its all parts are different from each other one can independently work on particular section of the project.
- Hence it is a separate section system the debugging and error solving in the MVC pattern is easy and fast.

The basic information about MVC component is mentioned here.

Model:

- Models are basically classes in c#.
- Model is nothing but the folder in our MVC project that keeps classes related to database management System. all the file which are responsible for accessing data from the database should be kept here.
- All the database related operations such as fetch data or update data are done in model
- It represents information about a domain that can be the application data of a web application.
- It means all the data of the application is stored in the Model.
- Model is like a bridge between application and database

How to add or create model in our project:

1. We have to create first MVC project in visual studio.
 2. Than
 3. solution explorer and search for the model folder.
 4. In this right click and add new class file .
 5. Create new class and click on add button.
 6. Write the code for the class file.
 7. Later the class will mapped to database as follows:
 8. The class name will be the table name in the database.
 9. Each member of the class will be mapped with table column with the same name.
 10. The datatype of this property will be the datatype of the column in the table.
-

View:

- The view component is used for all the UI logic of the application.
- For example ,the customer view all the UI components such as text boxes, dropdowns, etc.
- The end user GUI which user can interact with system like html,css,bootstrap is supported in the view.
- We can create different view for the application and rendered it on the local IIS server from the controller.
- It represent the presentation logic to provide the data of the logic.

Controller:

- Controller in MVC are also classes in c#. It extends the c# class named Controller Located in the System.web.mvc namespace.
 - It contain logic and provides link between model and view.
 - It represent the logic responsible for coordination between the view and the model classes.
 - It handles requests made by the user and then decides what to do according to the application.
 - It gets the data from the database (Model) and give it back to the view later the data is displayed by the view section.
 - It communicate with the model of the application and selects the required view to be redirect for the request.
 - It manages all the requests made by the user. And by doing that it manage the flow of the application.
 - It is a bridge between the model and view.
 - It contain all the logic,functions,method in short over all code required for the application.
 - It is responsible for locate the appropriate method to call for an incoming request.
 - It also handle the exception that the requested method throws.
 - Helps in rendering the view based on the result of the requested method.
-

Benifits of using MVC pattern:

- **Separation of concern:** provides the three main part for the view, model and controller.
 - **Simplified testing and maintance :** hence there is separate part for all the section testing of various fields are easy and also the maintance.
 - **Extensibility:** it is a separate system for all the concern so if we extend or update any part of the MVC pattern it will not affect the other.
-

3.Rest web API:

What is web API:

- API – application programming interface is a set of some subroutine definitions like functions, event, properties, etc and protocols which is used for communication and tools for building softwares.
- It is lightweight architecture and it can also be used with smartphone apps.
- API is one type of interface that has a set of predefined functions, methods etc. That allows programmer to access specific features or data of an application, on different types of the system like mobile, desktop, laptop etc.
- Web api as the name suggests is an API for web which can be accessed using HTTP protocol .
- It is created for the web applications. Like if some web application has one same module or feature for that we can create one web API and use that in all those app.
- It is all about resability.
- The ASP.net Web API is an extensible framework for building HTTP based services that can be accessed in the different applications on the different platforms.
- It will send data only as a response to the request. it only supports HTTP protocol.
- Web API can be used to access data from a database and save data back to the database.
- It uses the namespace System.web.Http.

What is restful services or restful API:

- REST or restful API (Representational State Transfer) is designed to take advantages of existing protocols.
 - REST is an architectural style or pattern to build web API .
 - In REST architecture , a REST server simply provides access to resources and the REST client accesses and presents the resources.
 - Here each resource is identified by URIs.
 - A resource is any information that has name like image, entity, document etc.
 - REST uses various representation to represent a resources like text, JSON and XML. JSON is now the most popular format being used in web services.
-

REST architectural pattern specifies a set of constraints that a system should follow or implement for creating restful services.

1. **Client Server:** the application should use client server model
 2. **Stateless:** it means it will not store the data into the server so the server will consume less memory.
 3. **Cacheable:** it can store the data which is requested by the user on the user machine and next time when the user request for that data it does not load from the server .it makes the process faster.
 4. **Uniform interface :** the application should have in such format that restachitecture supports. It discover the resource and the what http verb is used for communication between interface.
-

2. Start With Project

4. Create new web api project

- We can create web api project using asp .net web application template.
 - There are many templates available in visual studio for creating web api like empty template, API template etc. We can choose that as per our requirement.
 - Web api will always return only data does not return any view. Later client will decide how to display that data to user.
 - The data will be returned only in XML or JSON format.
 - XML or JSON are very versatile language that we can run that uri in any client machine it will return data. it will run on the different different client machine.
 - We can create Model , view and controller for the web api .the working is same as MVC. And also we are using some of the MVC functions.
 - We have to write the URI for the web api on our local host for the output.
 - Api/controller_name
 - It will run on the Google chrome browser and the server which is being used is IIS server.
 - We can also consume data from the created controller and use that in other file.
-

5. Create Controller and model:

Creating controller:

- We can create controller in web api as well as in MVC.
- The method is same for both but for the execution of the controller in mvc we hit the URI : `{controller}/{action}/{id}"`
- Controller name followed by action method which will be invoked is the sequence for the MVC Controller.
- here id is optional .it means if our controller method contain any parameter we can pass the id in the url and if it does not contain any parameter we can simply go with action method.
- For the web api the url for controller is different.
- `Api/{Controller}/{id}`
- We can create multiple methods in the controller and then they can be called by url

Creating Model:

- Models are basically classes in c# .
 - We can ass model classes that define database entities.then we can add web api controllers to perform CRUD operation on those entities.
 - The process of creating model in asp.net web application is same as creating controller.
 - We can add new class file by right clicking the model folder.
 - We can add class files as per our requirement.
 - The model class can be used in the view to populate the data, as well as sending data to the controller.
-

6. Passing parameter (from URL and from body)

- You can pass parameters to the web api controller method using either the [from body] or the [from uri] attribute .
 - If the parameter is a simple type ,web api tries to get the value from the URI
 - For complex types, web api tries to read the value from the message body, using media-type formatter.
 - So if we want to change the behaviour and force web API to read a complex type from the URI, add the [From Uri] attribute to the parameter.
 - To force web api to read a simple type from the request body , add the [FromBody] attribute to the parameter.
 - The request body is usually only used with CREATE or PUT method and often includes JSON object in body
-

7. Serialization:

- In c# serialization is the process of converting object into byte stream so that it can be saved in to memory , file or database or any persistence storage location.
- We can store object into memory with serilization and also retrieve it again when needed.
- The reverse of serialization is called deserialization.
- Together this process allows the data to be stored and transformed.
- 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 Iformatter interface which contain the methods serialize and de_serialize 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.

The namespace used for serialization is given here:

- System.Runtime.Serilization
- System.Xml.Serilization
- System.Text.Json

Advs of using serilization:

- Ability to transmit data across the network in cross-platform compatible format.
- Also for saving it in a persistent or non-proprietary format.
- Passing an object from one application to another.
- Passing an object through a firewall as an XML string.

.NET allows the following serilization techniques.

1. Binary serilization
 2. XML and SOAP serilization
 3. JSON serilization
-

Binary serialization:

- It is mechanism which writes the data to the output stream such as it can be used to re-construct the object automatically.
- Binary refers to that the necessary information that is required to create object is saved onto the storage media.
- It also preserves the instance identity.
- In other word the binary serialization the entire object state is saved where in XML only some of the object data is saved.

Advantages of binary serialization:

- It is faster and even more powerful in the sense that it provides support for complex object , read only properties.

Disadvantages :

- It is not easily portable to another platform.

XML serialization:

- It converts only the public fields and properties of object or parameters and return the values of methods into an XML stream.
- XML serialization result in strongly typed classes with public properties
- Fields that is converted in serial for storage purpose or transport purpose.
- Because XML is an open source standard , XML stream can be processed by any application as need.
- Implementing XML serialization in .NET is quite simple.
- The basic class we need is XmlSerializer for both serialization and de-serialization.
- It much slower compare to binary serialization.

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 to the JSON property names and copies the values for you.

There are 2 methods used for that :

1. JsonConvert : it is used for the simple scenario where you want to convert to and from a JSON string the serialize and deserialize methods on JsonConvert provide easy use wrapper over JsonSerializer.
2. JsonSerializer : the more control over how an object is serialized ,the JsonSerializer can be used directly.

The JsonSerializer is able to read and write JSON text directly to a stream via JsonTextReader and JsonTextWriter.

It has number of properties on it to customize how it serialize JSON.

8. Routing:

What is routing web API

- ASP.NET routing is a pattern matching system that is responsible for mapping incoming browser requests to specified controller action methods.
- In simple terms it is a specified route for the action method to be executed.
- It process the browser request and match that url with specified route and redirect user to that method.

How the routing is performed:

- When the request's URL matches any of the registered route pattern in the route table then the routing engine forwards the request to the appropriate handler for that request.
- thereafter the route is processed and gets a view on UI
- and when the request URL does not match any of the registered route patterns then the routing engine indicates that it could not determine the route registered for the requested URL.
- So basically when the routing engine receives a request at runtime it matches that request's URL with registered patterns and give the response accordingly.
- That's how it is work.

The default route for the route is :

url: api/{Controller}/{id}

types of routing:

1. attribute routing (static routing)
2. convention routing(default routing)

conventional routing:

- in this type of routing ,web API uses route template to determine which controller and action method to execute.
 - At least one route template must be added into route table to handle HTTP request.
 - When we create New web api project there is a appconfig folder in solution explorer. and in this folder default route is specified by the template for us.
 - If we will not provide any other route it will follow this route.
-

- The `config.Routes` is a table for all route collection .
- The method `MapHttpRequest()` is a method for mapping route with url we can add new route with using this method.
- We can add multiple routes in the web api using `HttpConfiguration` object.

Attribute routing:

- As the name suggest we have to write attribute **[route()]** to define routes.
 - It can be applied on any controller or action method.
 - When we have multiple action method in controller which take different parameters we can use attribute routing .
 - Because when we hit the url it will confuse that the multiple method to which to be executed in response. So here we can use attribute routing to avoid confusion in calling methods.
-
- If there are many methods in controller and all have some same pattern in beginning of the attribute routing we can define it above the controller so it will available for all methods .
 - we Have to write only the different route pattern in all methods.
 - There is also some special features like we can limit the attribute passing in the url.
 - Ex id should always greater or equal to 1 so we can give min limit attribute `m` also same for the max limit .then we can also give range attribute.
 - So these are some keyword that we can use along with route attribute.
-

9. Config :

- We can configure web API to customize the behavior of web API hosting infrastructure and component such as routes, formatters, filters etc.
 - Web API configuration process starts when the application starts.
 - In Asp.NET application, configure Web API by calling `GlobalConfiguration.Configure` in the `Application_Start` method.
 - The `configure()` method requires the callback method where Web API has been configured in code. By default this is the static `WebApiConfig.Register()` method.
 - `WebApiConfig.Register` method includes a parameter of `HttpConfiguration` type which is then used to configure the Web API.
 - The ASP.NET Web Application project template automatically sets up the configuration code.
 - The project template create a file named `WebApiConfig.cs` inside the `App_Start` folder.
 - This code file defines the delegates where we should put our Web API configuration code.
-