Day 7

Routing

- Mapping from http request to actions is knows as Routing.
- Routing will be done considering the controller name, action name, HTTP Verb (HTTP Get, HTTP Post) and any additional url parameters if any.

• Routing Engine & Routing Table

- Routing is handled by routing middleware or Routing Engine.
- In conventional routing, routing engine will create a routing table in the first running of the application.
- Routing table contains all possible routes to your application.
- Each time a user request comes to your application, routing table will be checked by routing engine.
- If the routing engine will not find a match then a status code 404 (Not Found) will be returned as output.

Handling Duplicate C# Method Signatures

- The the action parameter by default matches the C# method name on a controller.
- The ActionName attribute can be used to change in relation to routing.

```
[ActionName("Create")]
[HttpGet]
public IActionResult GetCar( ) {
```

```
[ActionName("Create")]
[HttpPost]
public IActionResult CreateCar( ) {
```

Custom Route Tokens

- In addition to the reserved tokens, routes can contain custom tokens that are mapped (model bound) to a controller action method's method's parameters.
- Eg: {controller}/{action}/{id?}

- Adding a question mark to a route token indicates that it is an optional route value and is represented in the action method as a nullable parameter.

Types of Routing

- Controller actions can be routed in 2 ways.
  - conventional routing
  - Attribute routing.

Routing

- Routing is how ASP.NET Core matches URL requests to controllers and actions (the endpoints)
- ASP.NET Core use the Routing middleware to match the URLs of incoming requests and map them to actions.

Conventional Routing

- Routing Table entries are defined in the Main Function/Top Level statements
- The MapControllerRoute() method adds an endpoint into the route table.

```
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Employ}/{action=stronglyTyped}/{id?}");
```

- While performing Routing, the routing table will be searched from top to find the first baching entry
- If the Http requests http method and the matching action's Http method matches then that action will be executed.

Attribute Routing

- In attribute routing, routes are defined using C# attributes on controllers and their action methods.
- Routes can contain hard-code values or can use token replacement
- Valid Tokens
  - [action] :- the action name where the route is defined
  - [controller] :- controller name where the route is defined

HTTP methods

- Controller methods can be decorated with the following attributes to indicate the HTTP method to respond to.
- [HttpGet] :
  - Is to respond to HTTP GET requests
  - Usually used to retrieve a resource

HTTP methods

- [HttpPost] :
  - Is to respond to HTTP POST requests
- [NonAction]
  - Indicates that a controller method is not an action method.

ViewBag, ViewData

ViewBag, ViewData

- The ViewBag, ViewData, and TempData objects are mechanisms for sending small amounts of data into a View.

ViewData	A dictionary that allows storing values in name/value pairs (e.g., ViewData["Title"] = "Foo").
ViewBag	Dynamic wrapper for the ViewData dictionary (e.g., ViewBag.Title = "Foo").

- Both ViewBag and ViewData point to the same object; they just provide different ways to access the data.

Difference between ViewBag and ViewData

- ViewData will do auto boxing while storing the values
- ViewBag will not do auto boxing.
- ViewBag will keep the values in its original type.