

## Form Tag Helpers in ASP.NET Core MVC || How to Create a Form in ASP.NET Core

.

Visual Studio interface showing a C# project named "Ex9\_Navigation".

**File Explorer (Solution Explorer):**

- Solution 'Ex9\_Navigation' (1 of 1 project)
  - Ex9\_Navigation
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
    - Controllers
      - HomeController.cs
      - Navigation.cs
      - StudnetFormController.cs (selected)
    - Models
    - Views
      - Home
      - Shared
        - \_Layout.cshtml
        - \_Layout1.cshtml
      - StudnetForm
      - \_ViewImports.cshtml
    - libman.json
    - Program.cs
    - Startup.cs

**Code Editor (Index.cshtml):**

```
1
2 @{}
3     ViewData["Title"] = "Index";
4     Layout = "~/Views/Shared/_Layout1.cshtml";
5 }
6
7 <h2>Student Forms</h2>
8 <form asp-controller="StudnetForm" asp-action="Index" method="post">
9     |
10    |
11    |
12 </form>
```

**Toolbox:** Empty.

**Status Bar:** 159 % zoom, No issues found.

**Taskbar:** Windows Start button, Search bar, and taskbar icons for File Explorer, Edge, and other applications.

- Asp-controller:--Pass the Controller name
- Asp\_action:--pass the action name
- Method:--post –to post the data on server.
- `<form asp-controller="StudnetForm" asp-action="Index" method="post">`
- `</form>`
- Run the Page
- `<form method="post" action="/StudnetForm">`
- It show the controller name only.
- `<form asp-controller="StudnetForm" asp-action="Create" method="post">`
- `</form>`
- If action name is Create it show
- `<form method="post" action="/StudnetForm/Create">`



Visual Studio interface showing the development of a web application named "Ex9\_Navigation".

**File Explorer (Solution Explorer):**

- Solution 'Ex9\_Navigation' (1 of 1 project)
  - Ex9\_Navigation
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
    - Controllers
      - HomeController.cs
      - Navigation.cs
      - StudnetFormController.cs
    - Models
      - Branch.cs
      - Gender.cs
      - Student.cs
    - Views
      - Home
      - Shared
        - \_Layout.cshtml
        - \_Layout1.cshtml
      - StudnetForm
        - \_ViewImports.cshtml
    - libman.json
    - Program.cs
    - Startup.cs

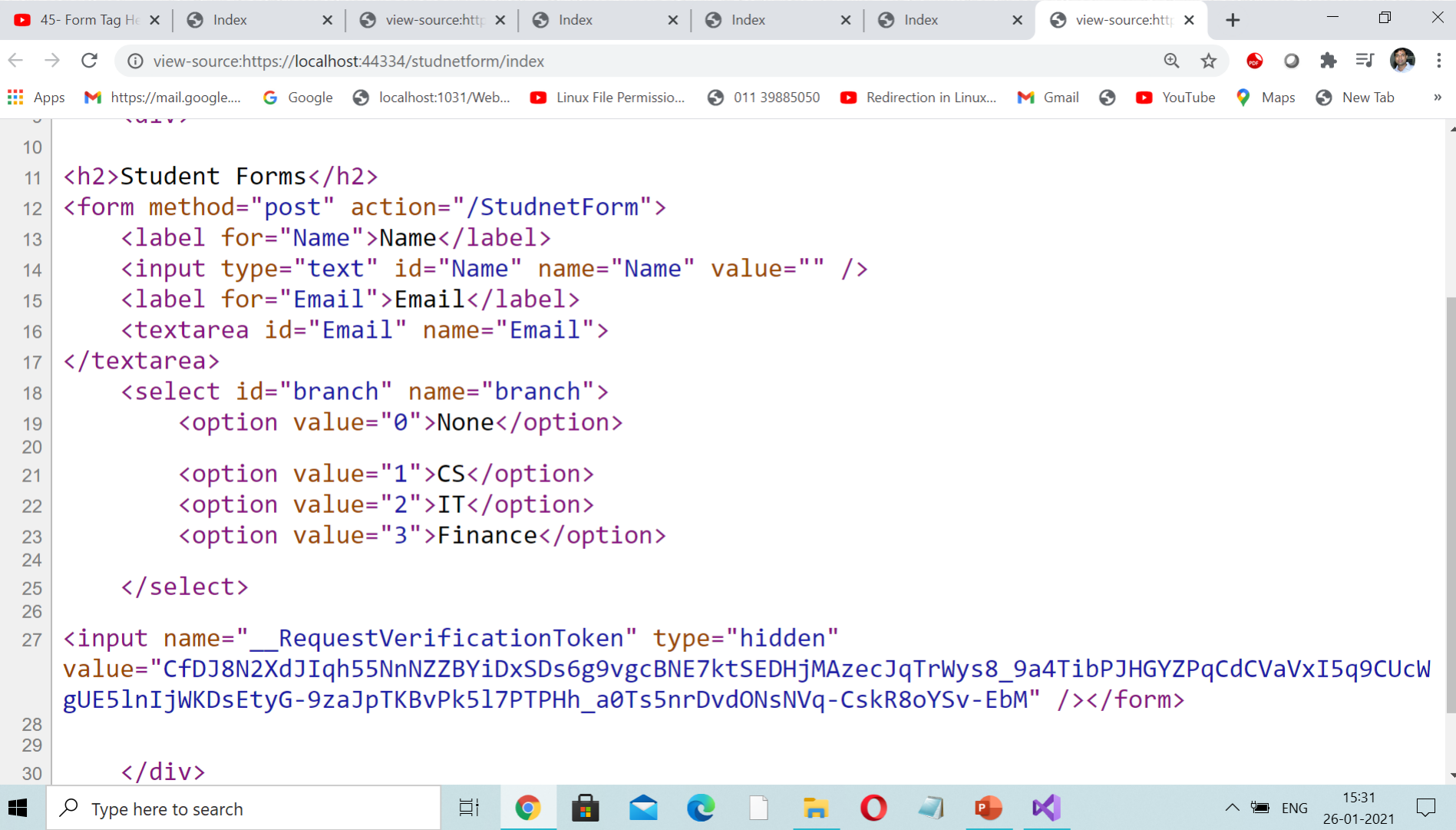
**Code Editor (Index.cshtml):**

```
6  
7 <h2>Student Forms</h2>  
8 <form asp-controller="StudnetForm" asp-action="Index" method="post">  
9     <label asp-for="Name"></label>  
10    <input asp-for="Name" />  
11    <label asp-for="Email"></label>  
12    <textarea asp-for="Email"></textarea>  
13    <select id="branch" name="branch">  
14        <option value="0">None</option>  
15  
16        <option value="1">CS</option>  
17        <option value="2">IT</option>  
18        <option value="3">Finance</option>  
19    </select>  
20  
21 </form>  
22  
23  
24
```

**Build Output:** Build succeeded

**System Tray:** 15:30, 26-01-2021

# After Run



45- Form Tag He... | Index | view-source:htt... | Index | Index | Index | view-source:htt...

view-source:https://localhost:44334/studnetform/index

```
10
11 <h2>Student Forms</h2>
12 <form method="post" action="/StudnetForm">
13     <label for="Name">Name</label>
14     <input type="text" id="Name" name="Name" value="" />
15     <label for="Email">Email</label>
16     <textarea id="Email" name="Email">
17 </textarea>
18     <select id="branch" name="branch">
19         <option value="0">None</option>
20
21         <option value="1">CS</option>
22         <option value="2">IT</option>
23         <option value="3">Finance</option>
24
25     </select>
26
27 <input name="__RequestVerificationToken" type="hidden"
28 value="CfDJ8N2XdJIqh55NnNZZBYiDxSDs6g9vgcBNE7ktSEdHjMAzecJqTrWys8_9a4TibPJHGYZPqCdCVaVxI5q9CUcW
29 gUE5lnIjWKDsEtyG-9zaJpTKBvPk5l7PTPhh_a0Ts5nrDvd0NsNVq-Cskr8oYSv-EbM" /></form>
30
31 </div>
```

Type here to search

15:31  
26-01-2021

- `<h2>Create</h2>`
- `<form asp-controller="StudentForm" asp-action="Create" method="post">`
- `<div class="form-group">`
- `<label asp-for="Name" class="col-sm-2 col-form-label">`
- `</label>`
- `<div class="col-sm-10">`
- `<input asp-for="Name" class="form-control" placeholder="Name" />`
- `</div>`
- `</div>`
- `<div class="form-group">`
- `<label asp-for="Email" class="col-sm-2 col-form-label">`
- `</label>`
- `<div class="col-sm-10">`
- `<input asp-for="Email" class="form-control" placeholder="Email" />`
- `</div>`
- `</div>`
- `<div class="form-group">`
- `<label asp-for="gender" class="col-sm-2 col-form-label">`
- `</label>`



- <div class="col-sm-10">
- @foreach (var gen in Model.gender)
- {
- <label>
- <input type="radio" asp-for="gender" value="@gen" id="gender@(gen)" />@gen
- </label>
- }
- </div>
- </div>
- <div class="form-group">
- <label asp-for="Address" class="col-sm-2 col-form-label">
- </label>
- <div class="col-sm-10">
- <textarea asp-for="Address" class="form-control" placeholder="Address"></textarea>
- </div>
- </div>
- <div class="form-group">
- <div class="col-sm-10">
- <button type="submit" value="Submit" class="btn btn-primary">Submit</button>
- </div>

- 
- How to Perform Model Binding in ASP.NET Core MVC



# What is Model Binding in ASP.NET Core?

- ▶ The Model Binding is a mechanism in ASP.NET Core Application which extracts the data from an HTTP request and provides them to the controller action method parameters.
- ▶ The action method parameters may be simple types like integers, strings, etc. or complex types such as Student, Order, Product, etc

- ▶ As we know it is our controller action method which is going to handle the incoming HTTP Request in ASP.NET Core MVC Application.

# HTTP Request Data Sources:

- ▶ ASP.NET Core MVC uses three primary data sources to map the HTTP requests data to the action method parameter in the following order:
- ▶ **Form values:** Values in the FORM in HTTP POST requests.
- ▶ **Route values:** Values provided by the Routing system.
- ▶ **Query string:** Values found in the URL's query string (e.g. after ? character).



# Pass the data using URL

Visual Studio interface showing the code for `EmployeeController` in the `Ex9_Navigation` project.

```
1 using Microsoft.AspNetCore.Mvc;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Threading.Tasks;
6
7 namespace Ex9_Navigation.Controllers
8 {
9     public class EmployeeController : Controller
10    {
11        public IActionResult Index()
12        {
13            return View();
14        }
15        public string Getname(int id, string name)
16        {
17            return "Id=" + id + "Name=" + name;
18        }
19    }
20 }
```

The Solution Explorer on the right shows the project structure:

- Solution 'Ex9\_Navigation' (1 of 1 project)
  - Ex9\_Navigation
    - Connected Services
    - Dependencies
    - Properties
    - wwwroot
    - Controllers
      - EmployeeController.cs
      - HomeController.cs
      - Navigation.cs
      - StudnetFormController.cs
    - Models
      - Branch.cs
      - Gender.cs
      - Student.cs
    - Views
      - Home
      - Shared
        - \_Layout.cshtml
        - \_Layout1.cshtml
      - StudnetForm
        - \_ViewImports.cshtml
    - libman.json
    - Program.cs
    - Startup.cs

Build succeeded. No issues found.

```
7 namespace Ex9_Navigation.Controllers
8 {
9     public class EmployeeController : Controller
10    {
11        public IActionResult Index()
12        {
13            return View();
14        }
15        public string Getname(int id, string name)
16        {
17            return "Id=" + id + "Name=" + name;
18        }
19        public string GetnameAge(int id, string name, int age)
20        {
21            return "Id=" + id + "Name=" + name + "Age=" + age;
22        }
23    }
24 }
25
```

46- How to Perform Model Binding x https://localhost:44334/Employee x +

localhost:44334/Employee/Getname/1?name=ss

Apps https://mail.google.... Google localhost:1031/Web... Linux File Permissio... 011 39885050 Redirection in Linux... Gmail YouTube Maps New Tab

Id=1Name=ss

Windows taskbar with search bar and application icons.

Type here to search

Taskbar icons: Chrome, File Explorer, Mail, Edge, Word, PowerPoint, Visual Studio Code, and system tray icons (network, volume, battery, language).

21:33 26-01-2021



46- How to Perform Model Bindin

https://localhost:44334/Employee

https://localhost:44334/Employee

+

localhost:44334/Employee/Getnameage/1?name=raj&&age=20

🔍

☆

PDF

🔄

⚙️

☰

👤

⋮

Apps

https://mail.google....

Google

localhost:1031/Web...

Linux File Permissio...

011 39885050

Redirection in Linux...

Gmail

YouTube

Maps

New Tab

»

Id=1Name=rajAge=20

🪟

🔍 Type here to search

📁

🌐

🖨️

✉️

🌀

📄

📁

🔴

📄

🌀

🎨

📄

📁

⤴️

🔋

ENG

21:37

26-01-2021

💬

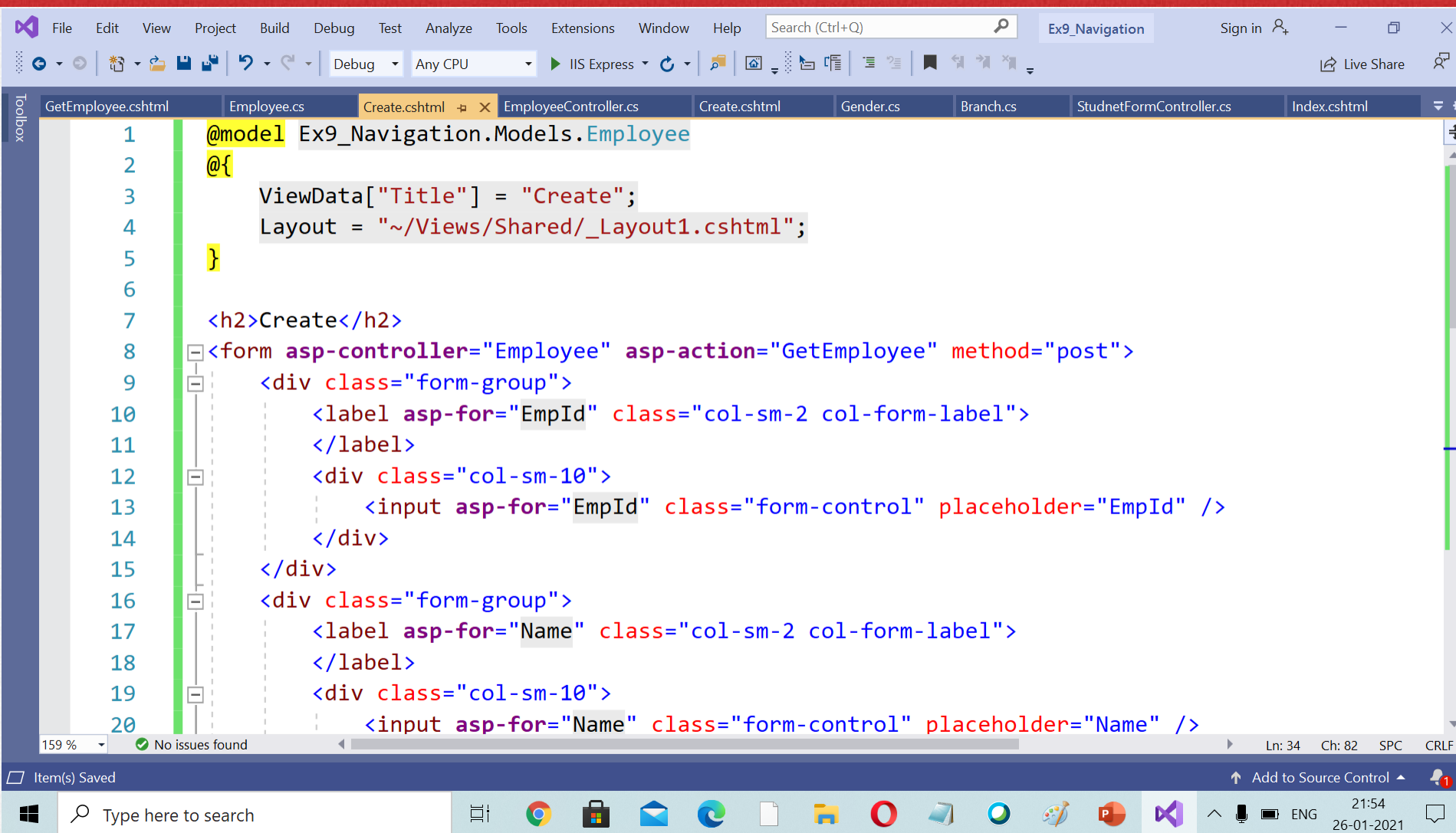
Visual Studio IDE showing the code for `EmployeeController.cs` in the `Ex9_Navigation` project.

The code defines the `EmployeeController` class, which inherits from `Controller`. It includes a `Create()` method and a `GetEmployee` method.

```
6 using Ex9_Navigation.Models;
7 namespace Ex9_Navigation.Controllers
8 {
9     public class EmployeeController : Controller
10    {
11        public IActionResult Create()
12        {
13
14            return View();
15        }
16        //if we are create same name of the method must you have to define http verbs.
17        //this method will be run time of post request data.
18        [HttpPost]
19        public string GetEmployee(Employee employee)
20        {
21            return "Id=" + employee.EmpId+"Name="+employee.Name+"EmailId="+employee.Emailid+"Ac";
22        }
23    }
24 }
```

The status bar indicates "Build succeeded" and "No issues found". The bottom of the image shows the Windows taskbar with the search bar and various application icons.

# Create asp-action="GetEmployee" to post the data in this action page.



The screenshot shows the Visual Studio Code editor with the following code in the `Create.cshtml` file:

```
1  @model Ex9_Navigation.Models.Employee
2  @{
3      ViewData["Title"] = "Create";
4      Layout = "~/Views/Shared/_Layout1.cshtml";
5  }
6
7  <h2>Create</h2>
8  <form asp-controller="Employee" asp-action="GetEmployee" method="post">
9      <div class="form-group">
10         <label asp-for="EmpId" class="col-sm-2 col-form-label">
11             </label>
12         <div class="col-sm-10">
13             <input asp-for="EmpId" class="form-control" placeholder="EmpId" />
14         </div>
15     </div>
16     <div class="form-group">
17         <label asp-for="Name" class="col-sm-2 col-form-label">
18             </label>
19         <div class="col-sm-10">
20             <input asp-for="Name" class="form-control" placeholder="Name" />
```

The interface includes a menu bar (File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help), a toolbar with icons for file operations and debugging, and a sidebar with a 'Toolbox' panel. The bottom status bar shows '159 %', 'No issues found', and 'Ln: 34 Ch: 82 SPC CRLF'. The Windows taskbar at the bottom displays the search bar and various application icons.



# Model Validation

- ▶ Sometimes the data entered by the user is not valid and cannot be saved into the database.
- ▶ The entered data may contain a typo or user may intentionally enter the inappropriate data.
- ▶ Hence, we need to validate the user input before storing it in the database
- ▶ The ASP.NET Core gives us Model Validator, which uses the validation attributes to validate the model, which makes our task easier.

# Model Validation

- ▶ The Form Data is posted to Controller action is automatically mapped to the action parameter by the Model Binder.
- ▶ The Model needs to be validated for the correctness. These validations can be done at the client side before sending data to the server or at the server side when the data is received from the client.
- ▶ The client-side validation is important because of better user experience, while the server side validation should ensure that the invalid data does not enter the system

# Validation Attributes

- ▶ The attributes applied on Model Properties are called **Validation attributes** and contain the code, which is used by the Model Validator to validate the model.
- ▶ The ASP.NET Core has lots of built-in Validation attributes, which can be added to the properties in a model, to enforce the validation rule.
- ▶ These Validation attributes also known as **DataAnnotations** and are available in [System.ComponentModel.DataAnnotations](#) namespace.
- ▶ You need to import it in your ViewModel to use them.



Thanks