

MVC.Net Core MVC

Handling Data

แบบ #1 - Controller

0 references

```
public IActionResult Form01()
{
    return View();
}
```

2

0 references

```
public IActionResult Form01Handle(string fname, string lname)
{
    ViewBag.fname = fname;
    ViewBag.lname = lname;
    return View();
}
```

แบบ #1 - View – Display Form

```
<!DOCTYPE html>

<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Index</title>
</head>
<body>
  <form method="post" asp-controller="Test01" asp-action="Form01Handle">
    First name: <input type="text" id="fname" name="fname" placeholder="John"><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname" placeholder="Doe" size="50"><br><br>
    <input type="submit">
  </form>
</body>
</html>
```

← → ↻ ⚠ Not secure | localhost:5001/Test01/Form01

First name:

Last name:

แบบ #1 – View – Submit

```
<!DOCTYPE html>

<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Form01Handle</title>
</head>
<body>
  <div>
    Name: @ViewBag.fname <br />
    Lastname: @ViewBag.lname
  </div>
</body>
</html>
```

แบบ #1 Conclusion

<http://localhost:5001/Test01/Form01>

← → ↻ ⚠ Not secure | localhost:5001/Test01/Form01

First name:

Last name:

```
rm method="post" asp-controller="Test01" asp-action="Form01Handle">
First name: <input type="text" id="fname" name="fname" placeholder="
<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname" placeholder="Doe" size="
<input type="submit">
```

```
0 references
public IActionResult Form01()
{
    return View();
}

0 references
public IActionResult Form01Handle(string fname, string lname)
{
    ViewBag.fname = fname;
    ViewBag.lname = lname;
    return View();
}
```

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Form01Handle</title>
</head>
<body>
  <div>
    Name: @ViewBag.fname <br />
    Lastname: @ViewBag.lname
  </div>
</body>
</html>
```

← → ↻ ⚠ Not secure | localhost:5001/Test01/Form01Handle

Name: John
Lastname: Smith

ပပပ #2 – Controller

0 references

```
public IActionResult Index()  
{  
    return View();  
}
```

[HttpPost]

0 references

```
public IActionResult Index(string fname, string lname)  
{  
    //ViewBag.Message = fname + " " + lname;  
    FLClass person = new FLClass();  
    person.FName = fname;  
    person.LName = lname;  
  
    return View(person);  
}
```

แบบ #2 – Model

2 references

```
public class FLClass
```

```
{
```

1 reference

```
public string FName { get; set; }
```

1 reference

```
public string LName { get; set; }
```

```
}
```

แบบ #2 – View

```
<!DOCTYPE html>

<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Index</title>
</head>
<body>
  <form method="post" asp-controller="Test01" asp-action="Index">
    First name: <input type="text" id="fname" name="fname" placeholder="John"><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname" placeholder="Doe" size="50"><br><br>
    <input type="submit">
  </form>
  <@if (ViewContext.HttpContext.Request.Method == "POST")>
  {
    <div>
      @*@ViewBag.Message*@
      @Model.FName @Model.LName
    </div>
  }
</body>
</html>
```

← → ↻ ⚠ Not secure | localhost:5001/Test01

First name:

Last name:

← → ↻ ⚠ Not secure | localhost:5001/Test01

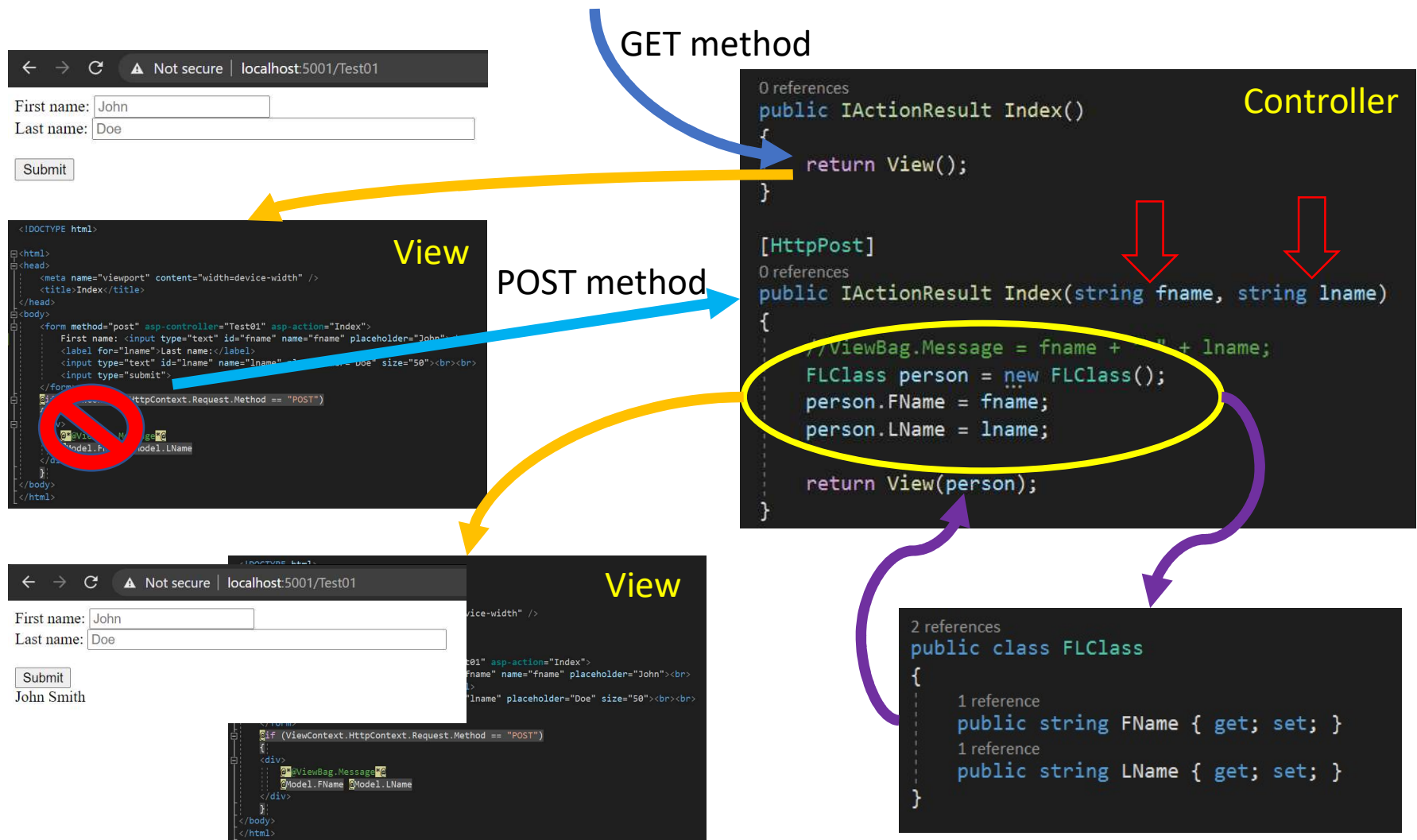
First name:

Last name:

John Smith

แบบ #2 – Conclusion

<http://localhost:5001/Test01/Index> or <http://localhost:5001/Test01>

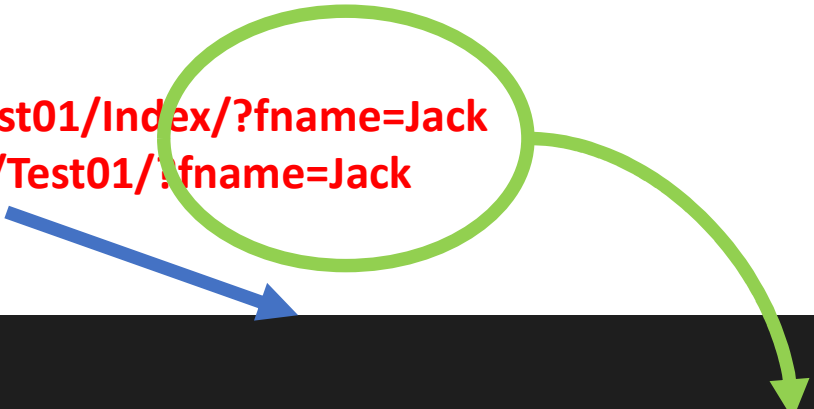


- แบบ **#1** และ แบบ **#2** สามารถใช้ผสมกันได้หรือสลับกันได้

GET Method data handling

Controller

<http://localhost:5001/Test01/Index/?fname=Jack>
or <http://localhost:5001/Test01/:fname=Jack>



A green circle highlights the parameter `fname=Jack` in the first URL. A blue arrow points from this circle to the `fname` parameter in the code. A green arrow points from the same circle to the `:fname` parameter in the second URL.

```
0 references  
public IActionResult Index(string fname)  
{  
    ViewBag.Message = fname;  
    return View();  
}
```

GET Method data handling

View

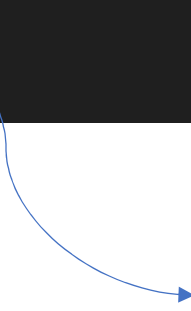
```
<html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>Index</title>
</head>
<body>
  <form method="post" asp-controller="Test01" asp-action="Index">
    First name: <input type="text" id="fname" name="fname" placeholder="John"><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname" placeholder="Doe" size="50"><br><br>
    <input type="submit">
    <div>@ViewBag.Message</div>
  </form>
</body>
</html>
```

← → ↻ ⚠ Not secure | localhost:5001/Test01/?fname=Jack

First name:

Last name:

Jack



Controller return

- View()
 - Web page -> html
 - View file in Views folder will be used to render response
- Content()
 - Can be anything that MIME supports

Example:

```
[Produces("text/plain")]
0 references
public IActionResult Index(string fname)
{
    //ViewBag.Message = fname;
    string x = "The received name is " + fname;
    return Content(x, "text/plain", Encoding.UTF8);
}
```

← → ↻ 🔒 localhost:5001/Test01/?fname=Jack

The received name is Jack

← → ↻ ⓘ view-source:https://localhost:5001/Test01/?fname=Jack

Line wrap ☐

1 The received name is Jack

Consume http

0 references

```
public IActionResult TestConsume()
{
    string output = "";
    HttpClient client = new HttpClient();
    client.BaseAddress = new Uri("https://localhost:5001/");
    var rtask = client.GetAsync("Test01/?fname=George");
    rtask.Wait();

    var result = rtask.Result;
    if (result.IsSuccessStatusCode)
    {
        var rcontent = result.Content.ReadAsStringAsync();
        rcontent.Wait();

        output = rcontent.Result;
    }
    return Content(output, "text/plain", Encoding.UTF8);
}
```