

ASP.Net Core MVC

Handling Data

2:56

แบบ #1 - Controller

Method สอง form + submit form

```
0 references
public IActionResult Form01()      Form แรก
{
    View 顯示 method นี้ ถ้า method นี้
    return View();  เป็น      Form01.cshtml
}
2
0 references
public IActionResult Form01Handle(string fname, string lname)
{
    ↗ ไป Global object 保存ค่าที่ได้รับ
    ViewBag.fname = fname;
    ViewBag.lname = lname;
    return View();
}
```

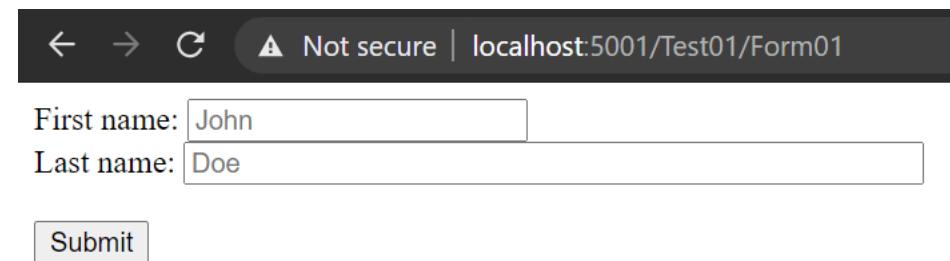
ต้องมีฟอร์มทุกหนึ่งหน้า
จะมี input ที่อยู่ใน form

หลังจาก submit

ແບ່ງ #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>
```



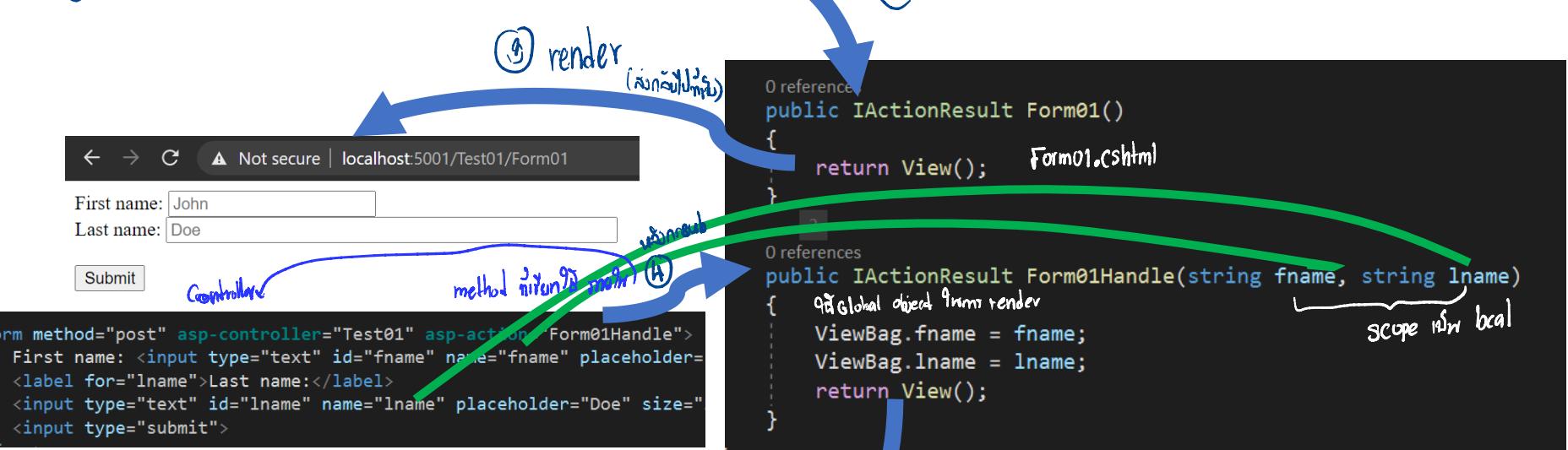
ແບ່ງ #1 – View – Submit

```
<!DOCTYPE html>

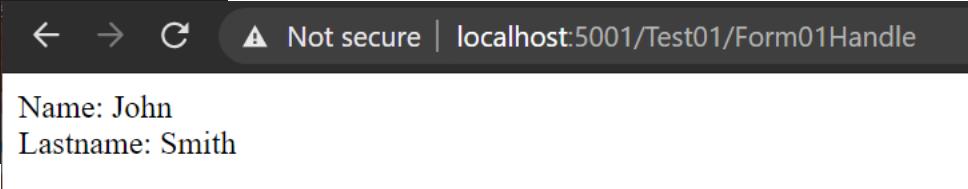
<html>
  <head>
    <meta name="viewport" content="width=device-width" />
    <title>Form01Handle</title>
  </head>
  <body>
    <div>      Data ໂກສດ໌
      |   Name: @ViewBag.fname <br />
      |   Lastname: @ViewBag.lname
    </div>
  </body>
</html>
```

กู้ๆ #1 Conclusion

① <http://localhost:5001/Test01/Form01>



```
<!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>
```



ເວັບແບບ #2 – Controller ອີ່ method ຕິດກຳກົດ display ໃນຂໍ້ມານີ້

```
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);
}
```

ກົດ
http request & target method ອີ່ໃຫຍ່ get
ກົດ
http request & target method (ອີ່ໃຫຍ່ post)
ກົດ
ກົດ default ອີ່ໃຫຍ່ get



5:09

عنوان #2 – Model

```
2 references
public class FLClass
{
    1 reference
    public string FName { get; set; }
    1 reference
    public string LName { get; set; }
}
```

3:20

לען #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: John

Last name: Doe

Submit

Click ①

① מילוי *

Not secure | localhost:5001/Test01

First name: John

Last name: Doe

Submit

John Smith ②

6:09

Controller → View
2 2 2 → [ViewBag
from class Model
(using)]

Lesson #2 – Conclusion

to Controller

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

(when Controller name didn't default = Run index)

Not secure | localhost:5001/Test01

First name: John

Last name: Doe

Submit

View

```
<!DOCTYPE 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" />
            Last name: <label for="lname"><input type="text" id="lname" name="lname" value="Doe" size="50" /><br><br><input type="submit" value="Submit" />
        </form>
        <div>@ViewBag.Message<br/>Model.FName & Model.LName</div>
    </body>
</html>
```

POST method

View

Not secure | localhost:5001/Test01

First name: John

Last name: Doe

Submit

John Smith

```
<!DOCTYPE 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" />
            Last name: <input type="text" id="lname" name="lname" value="Doe" size="50" /><br><br>
        </form>
        <div>@ViewBag.Message<br/>Model.FName & Model.LName</div>
    </body>
</html>
```

GET method

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);
}

View for Model contains person

Controller

Class

```
2 references
public class FLClass
{
    1 reference
    public string FName { get; set; }
    1 reference
    public string LName { get; set; }
}
```

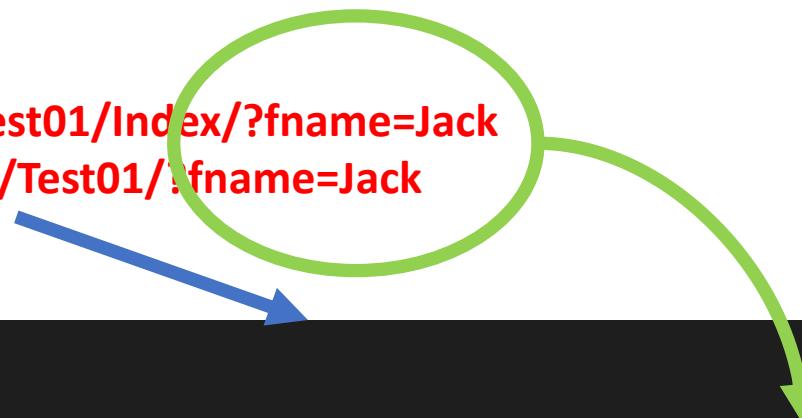
12.46

- แบบ #1 และ แบบ #2 สามารถใช้ผสานกันได้หรือสลับกันได้

GET Method data handling

Controller

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



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

GET Method data handling

View

The diagram illustrates the data flow from a C# code editor to a browser. A blue curved arrow originates from the '@ViewBag.Message' placeholder in the C# code and points to the 'Jack' message displayed in the browser's status bar.

```
<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>
```

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

First name: Last name:

Submit
Jack

Controller return

- View()
 - Web page -> html
 - View file in Views folder will be used to render response
from controller នៅពេលបង្កើតរាយការណ៍ នៃការសម្រេច (ជាដឹក html គ្នា។)
- Content()
 - Can be anything that MIME supports
 - Fixed type output
 - According to accept header of http message

23:33

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);
}

nếu muốn format trả về
localhost:5001/Test01/?fname=Jack
```

The received name is Jack និងពាណិជ្ជកម្មរបស់ខ្លួន, តើអាច tag html

```
← → ⌂ ⓘ view-source:https://localhost:5001/Test01?fname=Jack  
Line wrap   
1 | The received name is Jack
```

28:30

Consume http

```
0 references
public IActionResult TestConsume()
{
    string output = "";
    HttpClient client = new HttpClient();    9နိုတ်တော် URLများ
    client.BaseAddress = new Uri("https://localhost:5001/");
    var rtask = client.GetAsync("Test01/?fname=George"); လီယံးလုပ်ခွင့် Controller
    rtask.Wait(); ဘုရား server ဆိုလူသွေးသွေးလာ
                                                                controller
                                                                params
    var result = rtask.Result;  server ဆိုလူသွေးသွေးလာ (status code)
    if (result.IsSuccessStatusCode)
    {
        var rcontent = result.Content.ReadAsStringAsync();
        rcontent.Wait();  ကိုလိုက်ရှိစွာ = နောက်မှာ wait
        output = rcontent.Result;  စောင့်တာများ၏ text အမြဲ
    }
    return Content(output, "text/plain", Encoding.UTF8);
}
```

ASP.net

Ex 1. asp:Form

Ex. 6 107 form ex 1 method = POST 5 another way view model display