



A. MỤC TIÊU:

- Trình bày được cấu trúc của mô hình MVC
- Truyền dữ liệu với ViewBag, ViewData và Model

B. NỘI DUNG THỰC HÀNH

ViewData

Khái niệm:

ViewData là một thuộc tính của Controller base class, nó trả về một đối tượng ViewDataDictionary. ViewDataDictionary như tên của nó là một đối tượng dictionary cho phép lưu dữ liệu dạng key-value. Key phải là một chuỗi không phân biệt chữ hoa thường. Để truyền dữ liệu vào view bạn cần gán giá trị vào dictionary sử dụng key. Bạn có thể lưu bất kỳ số lượng key-value nào cần thiết trong ViewData.

ViewData truyền dữ liệu sang View từ Controller. Khi bạn gọi phương thức View trong Controller action, ViewData sẽ tự động gán vào View. Trong View bạn có thể truy cập giá trị được lưu trong ViewData cũng sử dụng key. Dữ liệu được lưu trong ViewData tồn tại chỉ trong request đó. Khi View được tạo xong cho client thì đối tượng ViewData đó cũng bị hủy.

Cách sử dụng

Trong ví dụ dưới đây nêu cách dùng ViewData trong Controller action:

```
public IActionResult SomeAction()
{
    ViewData["Greeting"] = "Hello World!";
    return View();
}
```

Trong ví dụ trên, chúng ta thêm chữ "Hello World" vào ViewData sử dụng key là "Greeting". Bạn có thể gọi phương thức dưới đây trong View để nhận giá trị từ key "Greeting":

```
@ViewData["Greeting"]
```

Gán đối tượng cho ViewData

Trong ví dụ trên, chúng ta lưu chuỗi dữ liệu vào ViewData. Chuỗi dữ liệu có thể được dùng trực tiếp mà không cần chuyển kiểu. Bạn có thể lưu bất cứ kiểu nào của dữ liệu như kiểu số nguyên, kiểu logic hay đối tượng trong ViewData.

Để sử dụng như một kiểu dữ liệu khác thì bạn cần phải chuyển kiểu giá trị từ ViewData sang kiểu tương ứng khi bạn dùng nó. Ví dụ dưới đây nêu cách dùng ViewData để gán đối tượng từ Controller sang View:

```
public IActionResult Index()
{
    ViewData["Greeting"] = "Hello World";
    ViewData["Product"] = new ProductModel()
    {
        ProductID=1,
    }
}
```

```

        Name = "Samsung galaxy Note",
        Brand = "Samsung",
        Price = 19000
    };
    return View();
}

```

Trong View, chúng ta lấy sản phẩm từ ViewData và chuyển sang kiểu ProductModel và sử dụng nó:

```

@{
    // Since Product isn't a string, it requires a cast.
    var product = ViewData["Product"] as ProductModel;
}
@ViewData["Greeting"]!

@product.ProductID<br>
@product.Name<br>
@product.Brand<br>
@product.Price<br>

```

Bạn có thể dùng ViewData để gán dữ liệu từ Controller vào View bao gồm cả Partial View và Layout.

ViewBag

Khái niệm:

ViewBag sử dụng kiểu động (dynamic) mà chúng ta đã có trong phiên bản C# 4.0. Nó là một vỏ bọc của ViewData và cung cấp thuộc tính động cho ViewData.

ViewBag có thể tiện dụng hơn để làm việc mà không cần chuyển kiểu. Sử dụng ViewBag như sau:

```

public IActionResult SomeAction()
{
    ViewBag.Greeting = "Hello";
    ViewBag.Product = new ProductModel()
    {
        ProductID = 1,
        Name = "Samsung galaxy Note",
        Brand = "Samsung",
        Price = 19000
    };
}

```

```
return View();
}
```

Sau đó bạn sử dụng trong View:

```
@ViewBag.Greeting World <br>

@ViewBag.product.ProductID<br>
@ViewBag.product.Name<br>
@ViewBag.product.Brand<br>
@ViewBag.product.Price<br>
```

ViewModel Trong Controller

.Phương thức View nhận model như một tham số và tự động gán vào ViewData.Model.

```
public IActionResult CorrectWay()
{
    Customer customer = new Customer();
    return View(customer);
}
```

Trong View

```
@using MVCCoreApp.Models;

@model Customer ← Declare the model to use

@{
    Layout = null;
}

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Method 1</title>
</head>
<body>

    @{
        <p>Id :@Model.CustomerID </p>
        <p>Name :@Model.Name </p>
        <p>Name :@Model.Address </p>
    }

</body>
</html>
```

Intellisense

1. Giới thiệu bài tập mẫu

Bài 1. Xây dựng ứng dụng Web theo mô hình MVC

– Xây dựng 1 danh sách các chuỗi tên trong Controller, sử dụng ViewBag chuyển tải dữ liệu trong danh sách vừa tạo sang View.

Trương Mạnh Hùng
Nguyễn Hải Yến
Trương Thị Khánh Uyên
Trương Nguyễn Quỳnh Anh

Hướng dẫn:

– Trong controller, xây dựng action chứa danh sách và sử dụng ViewBag để truyền dữ liệu từ controller sang View hiển thị.

```
public ActionResult MHList()
{
    List<string> danh sach = new List<string>();
    danh sach.Add("Trương Mạnh Hùng");
    danh sach.Add("Nguyễn Hải Yến");
    danh sach.Add("Trương Thị Khánh Uyên");
    danh sach.Add("Trương Nguyễn Quỳnh Anh");

    ViewBag.DuLieu = danh sach;
    return View();
}
```

– Trong View.cshtml, hiển thị dữ liệu vừa truyền với ViewBag.

```
<body>
<div>
    @{
        if (ViewBag.DuLieu != null)
        {
            foreach (var item in ViewBag.DuLieu)
            {
                @item <br>
            }
        }
    }
</div>
</body>
```

Bài 2. Xây dựng lớp Employee gồm các thuộc tính sau: Id, Name, Salary, Address. Khởi tạo giá trị mặc định cho Employee là thông tin một nhân viên. Hiển thị thông tin của nhân viên đó trên View bằng cách sử dụng ViewData.

Employee Detail

Employee ID: **101**

Employee Name: **Deepak**

Employee Salary: **35000**

Employee Address: **Delhi**

Hướng dẫn:

Model, xây dựng lớp Employee:

```
public class Employee
{
    public int ID{get;set;}
    public string Name {get;set;}
    public string Address { get; set; }
    public int Salary {get; set; }

    public Employee()
    {
        ID=101;
        Name = "Deepak";
        Salary = 35000;
        Address = "Delhi";
    }
}
```

Control:

```
public ActionResult Index()
{
    var emp = new Employee();
    ViewData["emp"] = emp;

    return View();
}
```

View:

```
@using MvcApplication1.Models;
```

```
@{
    ViewBag.Title = "Index";
    var viewDataEmployee = ViewData["emp"] as Employee; //need type casting
}
```

```
<h2>Employee Detail </h2>
```

```
<p> Employee ID: <b>@viewDataEmployee.ID </b> </p>
```


```
<p> Employee Name: <b>@viewDataEmployee.Name </b> </p>
```

```
<p> Employee Salary: <b>@viewDataEmployee.Salary </b> </p>
```

```
<p> Employee Address: <b>@viewDataEmployee.Address </b> </p>
```

Bài 3. Xây dựng nhân viên gồm các thuộc tính sau: Mã nhân viên, Tên nhân viên, Giới tính, Hình ảnh minh họa, khởi tạo giá trị mặc định cho 1 nhân viên với các thông tin (ID=101; Name = "Deepak"; GioiTinh = 0; HìnhAnh = "NV01.jpg");
Hiển thị thông tin của nhân viên đó trên View bằng cách sử dụng ViewModel.

Thông tin chi tiết của một nhân viên

Mã nhân viên	101
Tên nhân viên	Deepak
Giới tính	<input checked="" type="radio"/> Nữ
Hình minh họa	

Hướng dẫn:

Model, xây dựng lớp NhanVien:

```
public class NhanVien
{
    public int ID{get;set;}
    public string Name {get;set;}
    public int GioiTinh { get; set; }
    public string HinhAnh {get; set; }

    public NhanVien()
    {
        ID=101;
        Name = "Deepak";
        GioiTinh = 0;
        HinhAnh = "NV01.jpg";
    }
    public NhanVien(int ma, string ten, int gt, string h
}
```

Control:

```
public ActionResult show()
{
    var nv = new NhanVien();

    return View(nv);
}
```

View:

```
@model NhanVien

@{
    var gt= "Nam";
    if(Model.GioiTinh==0)
    {
        gt= "Nữ";
    }
}

<h2 style="color:red; background-color:black;" class="w-50">Thông tin chi tiết của một nhân viên </h2>

<table class="table w-50">
    <tr>
        <td> Mã nhân viên </td>
        <td> @Model.ID </td>
    </tr>
    <tr>
        <td> Tên nhân viên </td>
        <td> @Model.Name</td>
    </tr>
    <tr>
        <td> Giới tính </td>
        <td> @Html.RadioButton("GT", Model.GioiTinh,true) @Html.Label(gt)</td>
    </tr>
    <tr>
        <td> Hình minh họa </td>
        <td> </td>
    </tr>
</table>
```




2. Bài tập tại lớp

Bài 1. Xây dựng danh sách phòng ban gồm các thuộc tính sau: Mã phòng, Tên phòng. Khởi tạo giá trị mặc định danh sách các phòng. Hiển thị thông tin danh sách đó trên View bằng cách sử dụng ViewBag.

Danh sách các phòng ban

- Phòng giám đốc (P1)
- Phòng kế hoạch (P2)
- Phòng kế toán (P3)
- Phòng sản xuất (P4)
- Phòng kinh doanh (P5)

Bài 2. Xây dựng danh sách các cuốn sách gồm các thuộc tính sau: Mã sách, Tên sách, Giá, AnhBìa. Khởi tạo giá trị mặc định danh sách các quyển sách. Hiển thị thông tin danh sách đó theo mẫu sau bằng cách sử dụng ViewModel.

 <p>Lễ Nào Em Không Biết 64200 VNĐ CHON MUA</p>	 <p>Em là nhà 58800 VNĐ CHON MUA</p>	 <p>Khu vườn ngôn từ 71250 VNĐ CHON MUA</p>	 <p>Người truyền ký ức 49640 VNĐ CHON MUA</p>
 <p>Đạo tình 77000 VNĐ CHON MUA</p>	 <p>Yêu 52200 VNĐ CHON MUA</p>	 <p>Ký ả ảnh trắng 115500 VNĐ CHON MUA</p>	 <p>Cuộc sống không giới hạn 74000 VNĐ CHON MUA</p>

Bài 3. Thông tin của một sản phẩm bao gồm: Mã sách, tên sách, Giá, Hình ảnh, Mã loại, Mã nhà xuất bản. Trong đó, Loại sách gồm có thông tin mã loại, tên loại và Nhà xuất bản lưu trữ mã nhà xuất bản và tên nhà xuất bản như hình bên dưới. Xây dựng trang View hiển thị tất cả thông tin được truyền tải từ Controller thông qua ViewBag và ViewModel.

DANH SÁCH NHÀ XUẤT BẢN

Mã nhà xuất bản	Tên nhà xuất bản
NXB01	Hà Nội
NXB02	Thanh Niên
NXB03	Kim Đồng
NXB03	Bách Khoa

DANH SÁCH LOẠI SÁCH

Mã loại	Tên loại
L01	Sách giáo khoa
L02	Sách từ điển
L03	Truyện đại học
L04	Truyện tranh

DANH SÁCH SẢN PHẨM

Mã sách	Tên sách	Giá	Hình ảnh	Mã loại	Mã nhà xuất bản
S01	Toán 10 Nâng cao	15000	h1.png	1	1
S02	Ngữ Văn 11	2100	h2.png	1	1
S03	Từ điển 1000 từ	56000	h3.png	2	2
S04	Anh - Việt 500 từ	47000	h4.png	2	3
S05	Anh - Anh	120900	h5.png	2	4
S011	Cơ sở dữ liệu	34000	h11.png	3	2
S014	Doreamon	45000	h14.png	4	3