các phương pháp để truyền dữ liệu từ Action sang View

Có 3 cách:

- Sử dụng ViewData (Dictionary là một List of cloment/items)

+ trong 1 phần tử gồm: key/value(object)

+ x["key"] = value

+ lấy trị: value=x["key"]

+ Ví dụ (đưa table ra home)

->Index.cshtml

@{

ViewData["Title"] = "Home Page";

string titleMessage = Convert.ToString(ViewData["TitleMessage"]);

List<Person> listOfPersons = ViewData["ListOfPersons"] as List<Person>;

}

<h1>@titleMessage</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Person</th>

<th>Name</th>

<th>Address</th>

<th>Email</th>

</tr>

</thead>

<tbody>

@foreach (var item in listOfPersons)

{

<tr>

<td>@item?.PersonId</td>

<td>@item?.Name</td>

<td>@item?.Address</td>

<td>@item?.Email</td>

</tr>

}

</tbody>

</table>

-> HomeController

public IActionResult Index()

{

var dal = new PersonDAL();

var data = dal.List();

ViewData["TitleMessage"] = "List of Person";

ViewData["ListOfPersons"] = data;

return View();

}

- Sử dụng ViewBag (dynami)

**=> Index.cshtml**

@{

ViewData["Title"] = "Home Page";

// string titleMessage = Convert.ToString(ViewData["TitleMessage"]);

// List<Person> listOfPersons = ViewData["ListOfPersons"] as List<Person>;

}

<h1>@ViewBag.TitleMessage</h1>

<table class="table table-bordered">

<thead>

<tr class="bg-primary">

<th>Person</th>

<th>Name</th>

<th>Address</th>

<th>Email</th>

</tr>

</thead>

<tbody>

@foreach (var item in ViewBag.ListOfPersons)

{

<tr>

<td>@item?.PersonId</td>

<td>@item?.Name</td>

<td>@item?.Address</td>

<td>@item?.Email</td>

</tr>

}

</tbody>

</table>

**=> HomeController**

using \_20T1080053.Web.Models;

using Microsoft.AspNetCore.Mvc;

using System.Diagnostics;

namespace \_20T1080053.Web.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

var dal = new PersonDAL();

var data = dal.List();

//ViewData["TitleMessage"] = "List of Person";

//ViewData["ListOfPersons"] = data;

ViewBag.TitleMessage = "List of Person";

ViewBag.ListOfPersons = data;

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

**=>>>>>>>**

\* Ư điểm: Sướng, khỏe

\* Nhược điểm: ko chặt chẽ => khổ về sau

- Sử dụng Binding Model (Strong typed data)

+ Sử dụng để truyền các dữ liệu có cấu trúc chặt chẽ

+ IEnumerable<Person> => Tập hợp nhiều

+ List<Person> => Chỉ một

**\* Lưu ý**

- Muốn đổi tên khai báo get, set ở class thì phải sử dụng ctrl + R để đổi, ko được xóa và đổi (sẽ lỗi)

- Truyền dữ liệu đầu vào cho Action (truyền tham số cho hàm)

+ Thực hiện thông qua Request (URL)

+ Có 2 cách để truyền:

++ Truyền thông qua tham số trên URL:

Ví dụ: <http://localhost:61717/home/getEmployee?name=PhongBage=40>

Tham số truyền dưới dạng Query String

Thường dùng khi gọi Action với phương thức GET

Nên gán giá trị mặc định cho các tham số để tránh lỗi

++ Truyền tham số thông qua body

Thường thông qua các form nhập dữ liệu sử dụng phương thức là POST

- Tham số đầu vào của Action thay vì sử dụng các tham số khai báo biến theo kiểu

- Liệt kê có thể sử dụng Model => Linh hoạt, dể refactor code

**Pantial**

**ViewConpunts**

Những tham số có tên lặp lại khi truyền thì hiểu là mảng

<http://localhost:61717/home/getEmployee?name=Phong&name=Nga&name=Hien&age=23>