Truyền data từ Controller sang View

# Mục tiêu bài thực hành

* Áp dụng ViewBag vào việc truyền data từ Controller sang View
* Áp dụng ViewData vào việc truyền data từ Controller sang View
* Phân biệt được sự khác nhau khi sử dụng ViewBag và ViewData.

Yêu cầu 1: Tạo project ASP.NET WebApplication theo template Empty / MVC

Yêu cầu 2: Trong project vừa tạo hãy tạo 1 controller có tên là Home trong thư mục Controller: HomeController.cs

# Sử dụng ViewBag

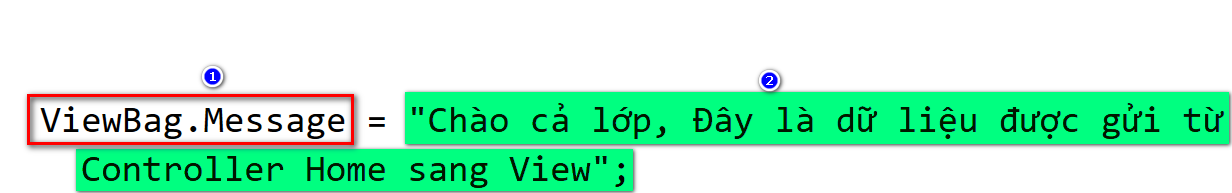
* Với kiểu dữ liệu build in
* Với kiểu Object
* Với kiểu List<Object>

Yêu cầu 3: Tạo View cho Action index của controller Home, có sử dụng layout mặc định của ASP.Net MVC

Yêu cầu 4: Hãy thực hiện khai báo để tiến hành truyền 1 dòng chữ sau sang view từ Action index của Controller Home. Nội dung: “Xin chào lớp, Dữ liệu này được truyền từ Controller Home sang View”

Hướng dẫn:

* Trong Action index của controller Home nhập khai báo ViewBag:



* Để hiển thị được đoạn văn bản trên view Index.cshtml. Trong View index hãy nhập đoạn code sau:



Trong đoạn coded này sử dụng ViewBag.Message được khai báo trong Controller home.

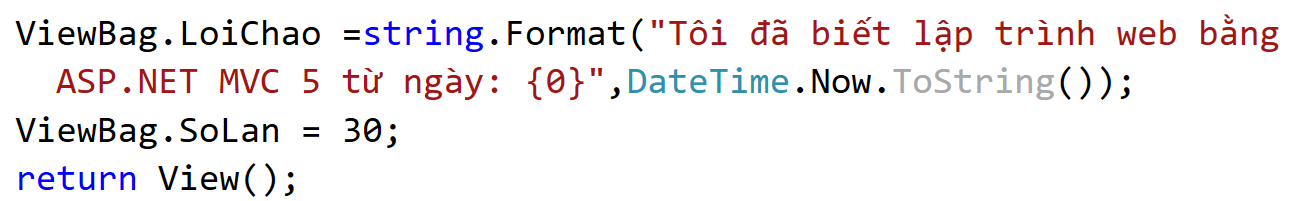
Nhận xét:

* ViewBag.Message được sử dụng như 1 biến được khai báo và gán giá trị trong Action của Controller.
* Khi sử dụng Biến này trong view, ta sử dụng cú pháp Razor để hiển thị (đánh sau dấu @)
* Khi sử dụng viewBag, ta gán giá trị kiểu gì cho ViewBag thì ViewBag sẽ tự động nhận kiểu dữ liệu đó mà không phải ép kiểu tường minh.
* Sau khi hoàn tất yêu cầu 4 hãy build project website để xem kết quả.

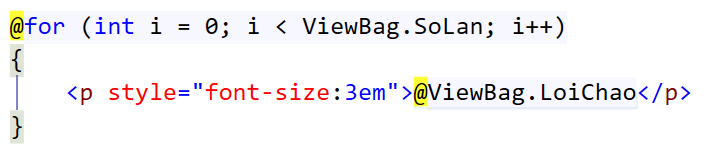
Yêu cầu 5: Tương tự yêu cầu 4 trong Action index của controller home. Hãy hiển thị dòng chữ “Tôi đã biết lập trình web bằng ASP.NET MVC5”. 30 lần trên view index.cshtml, với size chữ lớn=4em.

Hướng dẫn:

* Khai báo ViewBag trong Action Index của Controller Home.



* Trong view Index.cshtml hãy nhập đoạn code sau:



Trong đó:

+ Sử dụng @ để nhúng code C# (vòng lặp for theo C#) vào trong file HTML theo cú pháp của Razor.

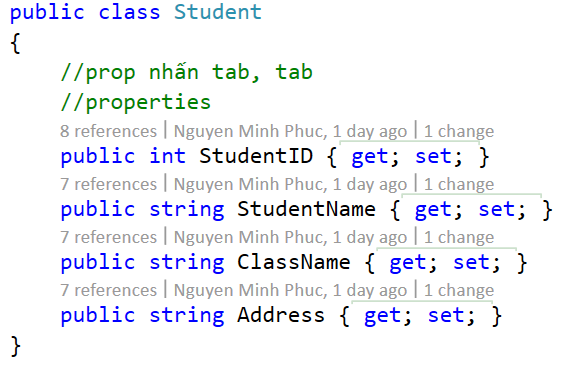
+ ViewBag.SoLan và ViewBag.LoiChao được lấy từ Action Index trong HomeController.

Yêu cầu 6: Sử dụng kiểu dữ liệu Object với ViewBag.

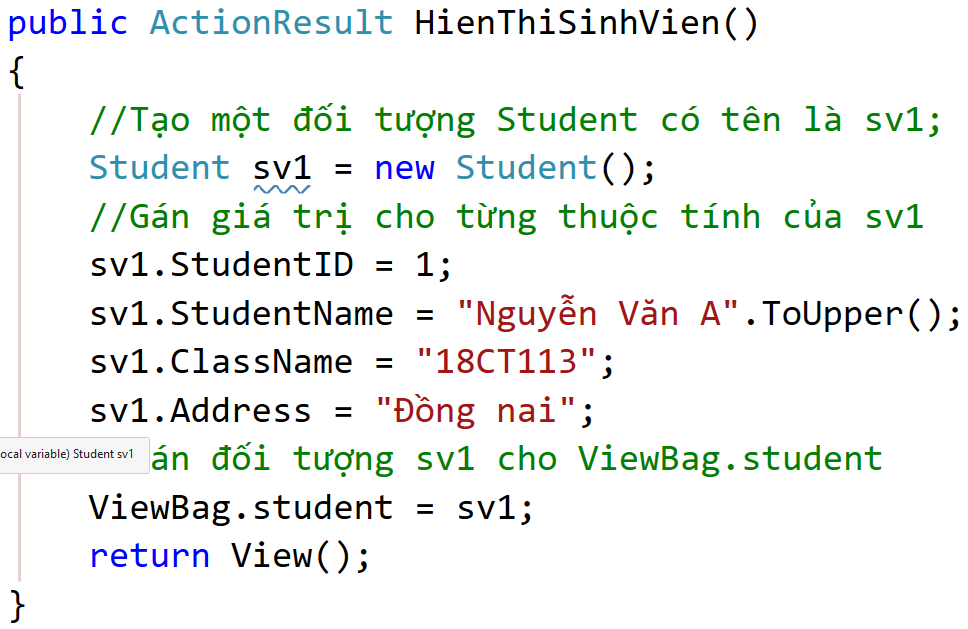
* Tạo một object Student gồm các thông tin sau: StudentID, StudetnName, ClassName, Address.
* Khởi tạo giá trị 1 student từ kiểu dữ liệu mới tạo
* Truyền đối tượng sinh viên sang view và hiển thị thông tin đó trên trình duyệt.

Hướng dẫn:

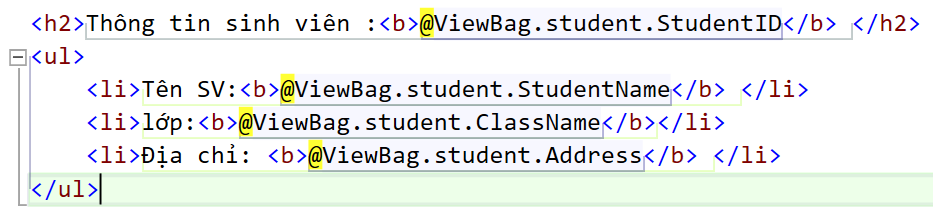
* Tạo một Class Student trong thư mục Models của Project website.
* Trong Class Student hãy khai báo các thuộc tính như yêu cầu



* Tạo một Action có tên là HienThiSinhVien() trong HomController.
* Nhập code khởi tạo giá trị cho các thuộc tính của Student.



* Tạo view HienThiSinhVien và tiến hành hiển thị thông tin sinh viên trên views.



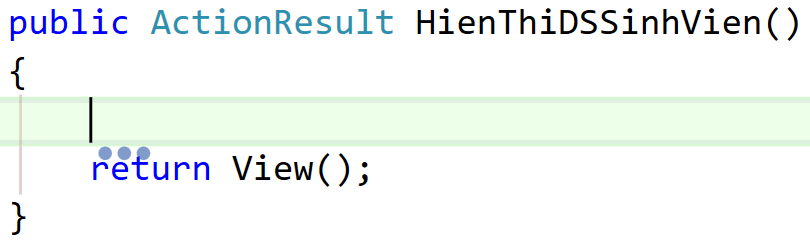
* Sau khi hoàn tất hãy chạy để xem kết quả trên trình duyệt.

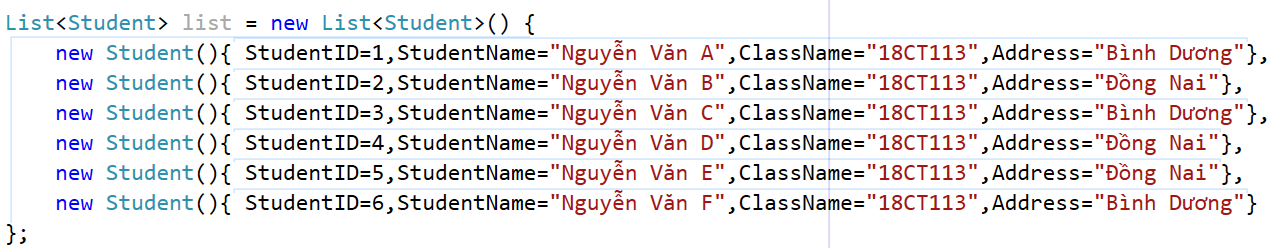
Yêu cầu 7: Sử dụng kiểu danh sách (List<Student>) với ViewBag

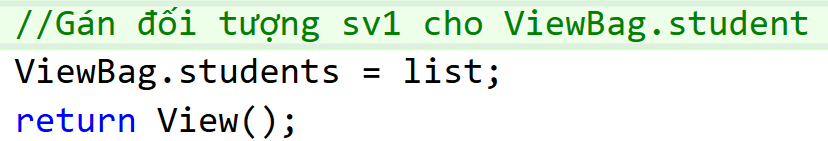
* Khởi tạo một danh sách student và hiển thị danh sách đó trên trình duyệt

Hướng dẫn:

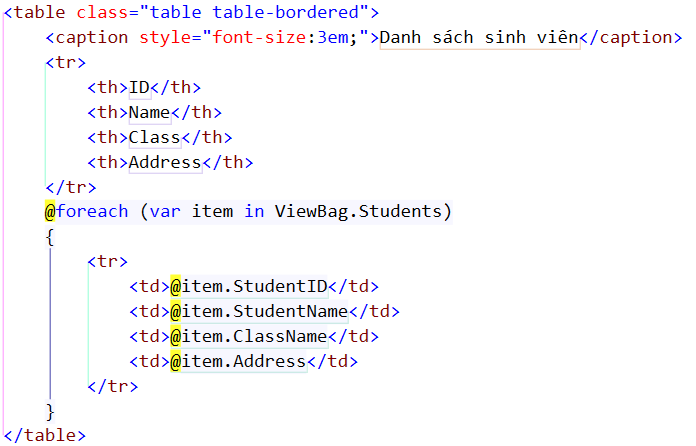
* Tại HomeController hãy tạo một Action có tên là HienThiDSSinhVien()



* Tạo đối tượng danh sách sinh viên List<Student> và nhập 1 danh sách 6 Student cho danh sách đó
* 
* Gán danh sách đó vào ViewBag

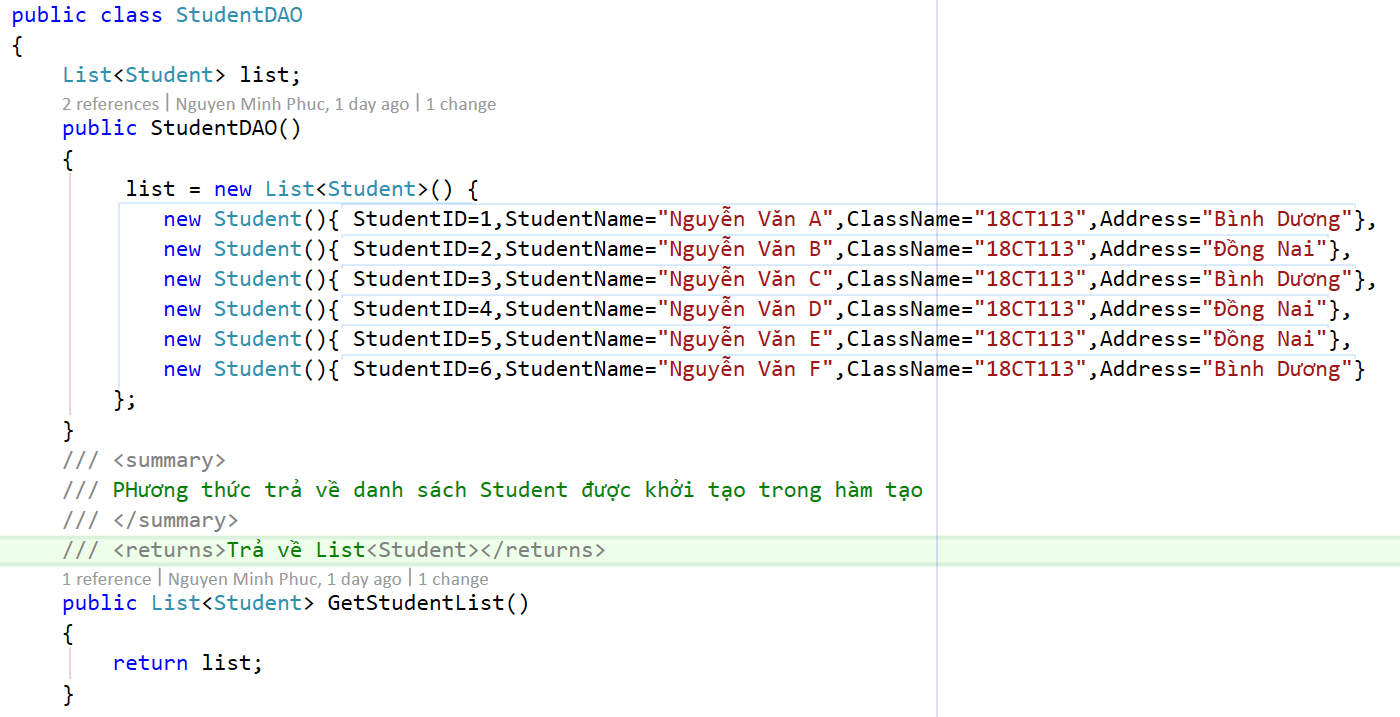


* Tạo view cho Action và HIển thị danh sách trên view: Sẽ hiển thị danh sách student trong table.

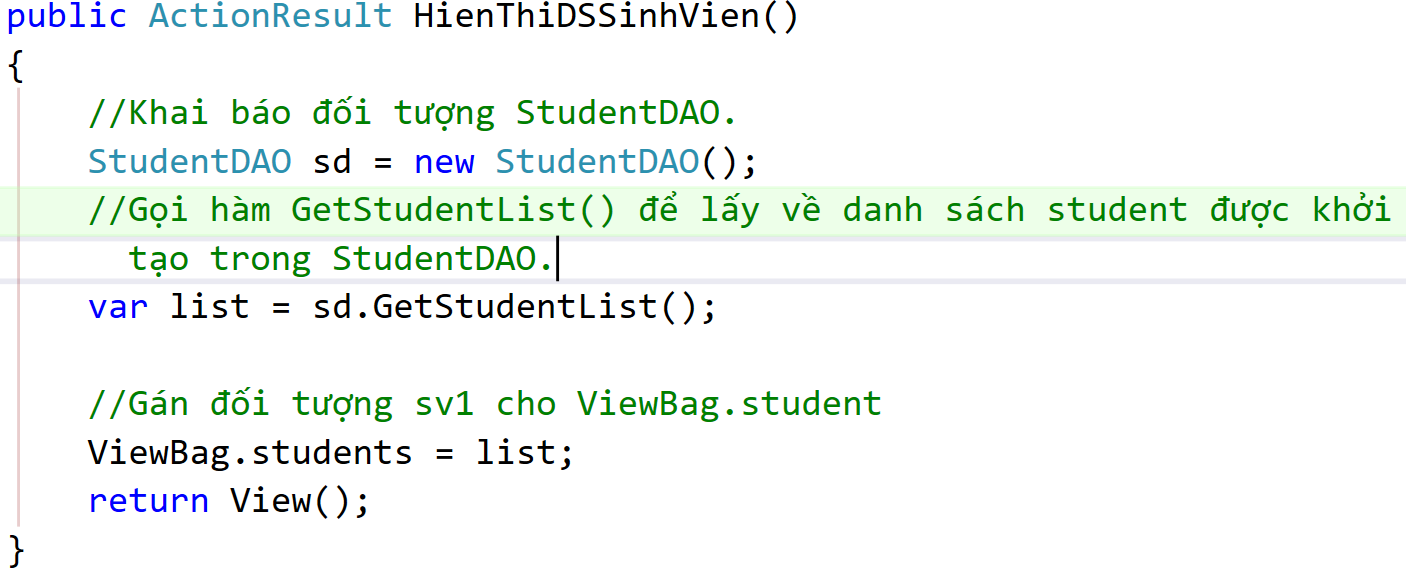


**Thực hiện yêu cầu trên có một cách khác như sau:**

* Thay vì khỏi tạo giá trị cho danh sách sinh viên trong Controller ta có thể tạo 1 class trong Models để thực hiện việc khỏi tạo giá trị cho danh sách ( sau này khi có kết nối đến database, việc lấy dữ liệu vào danh sách sẽ là phương thức đọc data từ Database)
* Tiến hành như sau:
  + Tạo một Class có tên là StudentDAO trong thư mục Models của project
  + Trong Class StudentDAO thực hiện khai báo như sau



* + Trong Controller tiến hành khai báo đối tượng StudentDAO để gọi data cho danh sách sinh viên (List<Student>).
  + Gán danh sách vào ViewBag

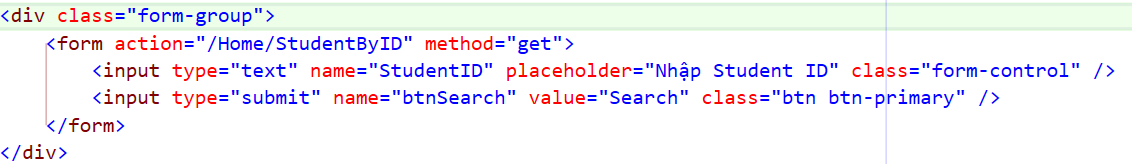


* + Tạo view để hiển thị (không thay đổi)

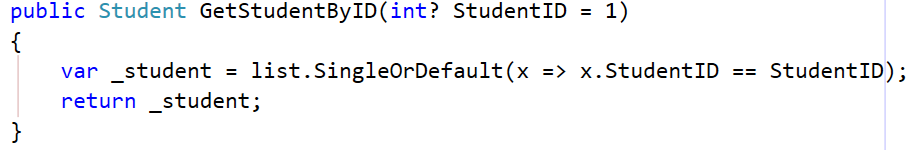
Yêu cầu 08: Từ view HienThiDSSinhVien hãy tạo 1 input text để thực hiện chức năng. Nhập mã sinh viên và chọn search sẽ cho hiển thị sinh viên đó trong 1 view mới, nếu không tìm thấy sẽ hiển thị sinh viên không có trong danh sách.

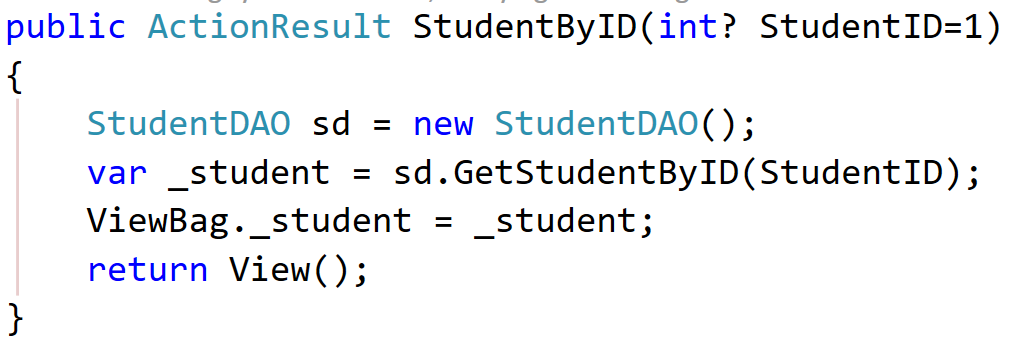
Hướng dẫn:

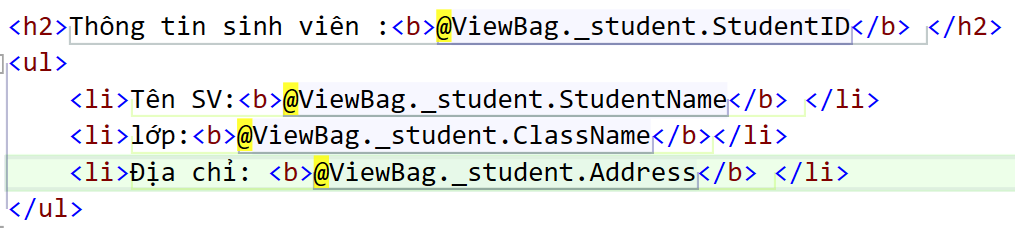
* Tại View HieenThiDSSinhVien tạo Form để nhập text tìm kiếm (Sử dụng URLParameter để truyền)



* Trong ClassStudentDAO hãy viết thêm phương thức GetStudentByID(int StudentID)



* Tạo 1 Action trong HomeController có tên là GetStudentByID (int StudentID) 
* Tạo 1 view cho Action GetStudentByID để hiển thi thông tin sinh viên tìm thấy



# Sử dụng ViewData

Sử dụng viewData giống như sử dụng viewBag tuy nhiên cách thức khai báo và cách sử dụng sẽ khác nhau

Sự khác nhau giữa ViewBag và ViewData

|  |  |  |
| --- | --- | --- |
| Đối tượng | ViewBag | ViewData |
| Khai báo Kiểu dữ liệu built in | ViewBag.Message=”String”  ViewBag.Number=100 | ViewData[“Message”]=”String”  ViewData[“Number”]=100 |
| Khai báo với kiểu đối tượng | ViewBag.Student=sv1 | ViewData[“Student”]=sv1 |
| Cách sử dụng | @ViewBag.Message  @ViewBag.Number  @ViewBag.Student.StudentID | @ViewData[“Message”]  Convert.ToInt32(@ViewData[“Number”]  @((ViewData[“Student”] as Student).StudentID) |

**Chú ý:** Khi sử dụng ViewData phải ép kiểu tường mình về đúng kiểu dữ liệu mà mình muốn sử dung.

Thực hành: Tương tự bài tập phần ViewBag. Hãy xây dựng một kiểu dữ liệu product chữa các thông tin sau: ProductID (int), ProductName (string), Price (double)

# Sử dụng TempData

* Cách khai báo và cách dùng ptương tự như ViewData
* Khi sử dụng cần Ép kiểu tường minh
* TempData còn để truyển data quan lại giữa các Controller với nhau
  + Tempdata có thể lưu dữ liệu qua 2 request
  + Muốn thực hiện nhiều hơn 2 lần hãy sử dụng hàm TempData.Keep()

Yêu cầu: