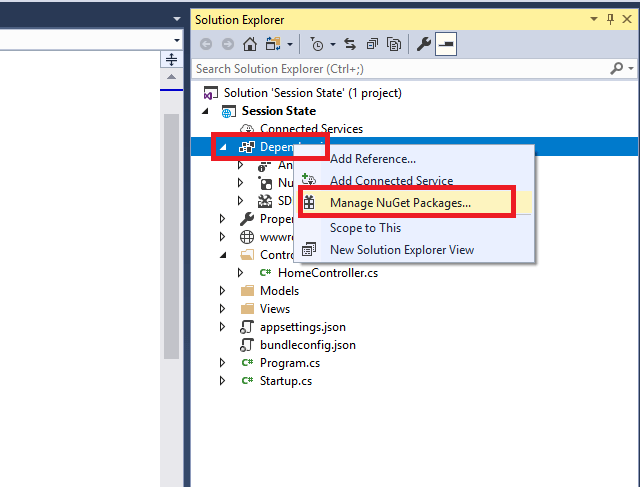
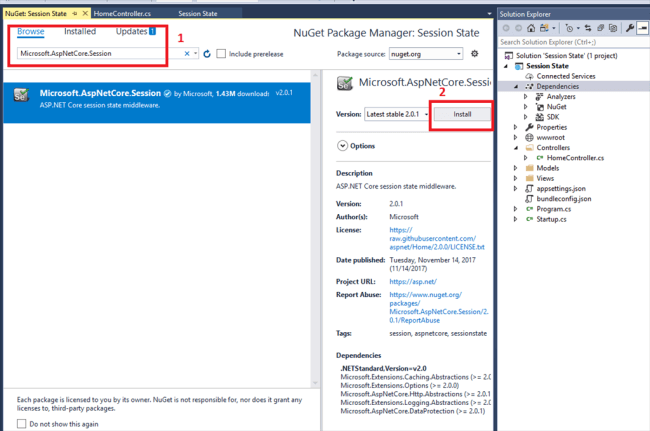
1. **Session**

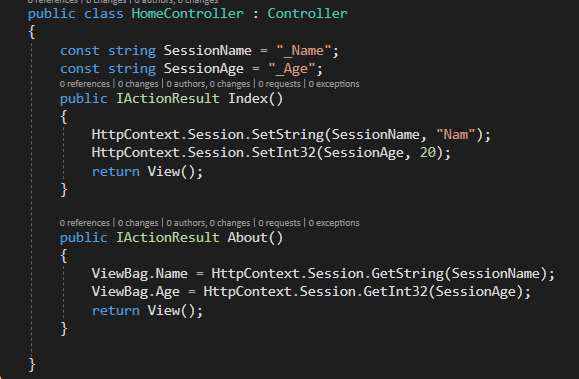
Step 1:



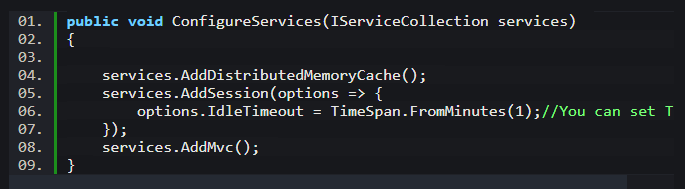
Step 2:



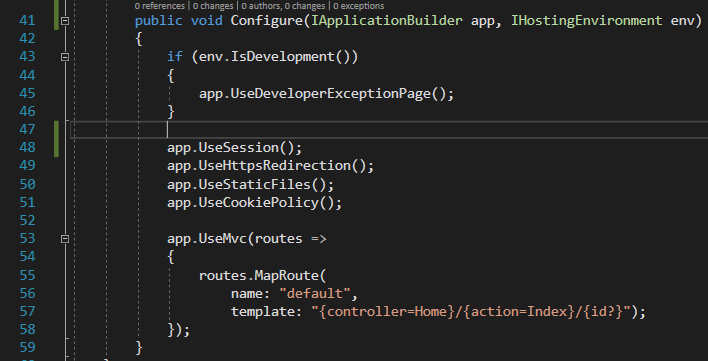
Step 3: edit in Controller



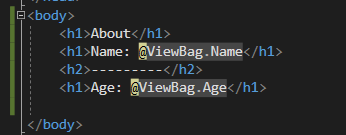
Step 4: Set a short timeout for easy test



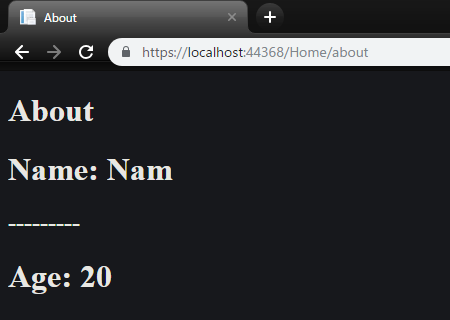
Step 5:



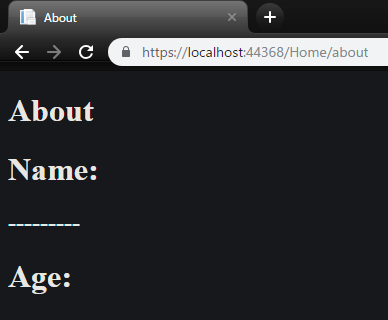
Step 6: Edit html



Step 7: Run app



Step 8: After 1 minutes, session will end



1. **Cookie ( Authentication and Authorization )**

Step 1:

1. **TempData, ViewBag, ViewData**

**ViewData**và **ViewBag**trong ASP.NET Core là những tập hợp dữ liệu dạng weak types (hay còn gọi là loose types), tức là chúng ta không cần định nghĩa trước kiểu dữ liệu một cách rõ ràng. Một số trường hợp sử dụng **ViewBag**và **ViewData**:

|  |  |
| --- | --- |
| Truyền dữ liệu | Ví dụ |
| Controller và View | Truyển dữ liệu vào DropdownList |
| View và Layout View | Thiết lập lại nội dung của element **<title>** trong Layout View từ một View con. |
| [Partial View](https://docs.microsoft.com/en-us/aspnet/core/mvc/views/partial) và View | Một Widget hiển thị dữ liệu phụ thuộc vào trang mà người dùng truy cập. |

Vì là weak types nên **ViewBag**và **ViewData**chỉ được xử lý lúc runtime và sẽ không có kiểm tra kiểu dữ liệu lúc compile-time như viewmodel nên sẽ dễ bị lỗi hơn. Do đó chúng ta chỉ nên sử dụng chúng khi cần truyền một lượng dữ liệu nhỏ và dễ kiểm soát.

Chú ý là ViewBag không sử dụng được trong Razor Pages.

* 1. ViewData

Với kiểu dữ liệu dạng **string**thì chúng ta có thể lưu trữ và sử dụng trực tiếp, còn với kiểu dữ liệu dạng **object**thì khi sử dụng chúng ta sẽ phải ép kiểu (cast) sang kiểu dữ liệu xác định. Ví dụ khi truyền một object từ Controller sang View:

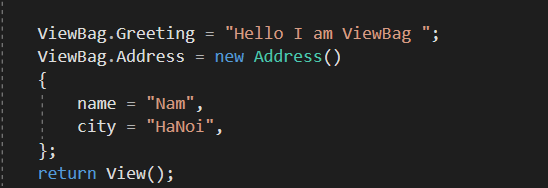


Sử dụng ViewData bên View:

* 1. ViewBag

Là một [DynamicViewData](https://docs.microsoft.com/en-us/dotnet/api/Microsoft.AspNetCore.Mvc.ViewFeatures.Internal.DynamicViewData?view=aspnetcore-2.0) object, nó là một lớp bao bọc (wrap) ViewData để cho phép truy cập vào object một cách linh hoạt. ViewBag cũng cho phép chúng ta sử dụng dynamic properties (dùng dấu chấm thay vì ngoặc vuông như ViewData). Sử dụng ViewBag cũng tương tự như ViewData nhưng sẽ tiện lợi hơn vì nó không cần phải ép kiểu. Ví dụ:

* 1. TempData



Sử dụng ViewBag bên View:

Tương tự [ViewData và ViewBag](https://techmaster.vn/posts/34555/so-sanh-viewdata-va-viewbag-trong-aspnet-core), TempData cũng dùng để truyền dữ liệu ra view. Tuy nhiên sẽ hơi khác một chút, đó là TempData sẽ tồn tại cho đến khi nó được đọc. Tức là ViewBag và ViewData chỉ hiển thị được dữ liệu ngay tại trang người dùng truy cập, còn TempData có thể lưu lại và hiển thị ở một trang sau đó và nó chỉ biến mất khi người dùng đã "đọc" nó.

TempData thường được ứng dụng để hiển thị các thông báo thành công, thất bại ở trang kế tiếp.

**Ví dụ với ứng dụng Quản lý Book có 2 trang Danh sách Book và Tạo mới Book.**

Khi người dùng Tạo mới Book xong sẽ được chuyển qua trang Danh sách Book kèm một thông báo tạo book thành công. Lúc này chúng ta sẽ không thể dùng ViewBag hay ViewData để lưu thông báo được vì thông báo nằm ở trang khác (Action khác). Chúng ta sẽ dùng TempData để lưu thông báo ở trang Tạo mới Book và hiển thị ra ở trang Quản lý Book:

Step 1: