[HttpGet]

public IActionResult Create()

{

ViewBag.Categories = new SelectList(\_dataContext.Categories, "Id", "Name");

ViewBag.Brands = new SelectList(\_dataContext.Brands, "Id", "Name");

return View();

}

[HttpPost] // Chỉ nhận POST requests

public async Task <IActionResult> Create(ProductModel product)

{

ViewBag.Categories = new SelectList(\_dataContext.Categories, "Id", "Name" , product.CategoryId);

ViewBag.Brands = new SelectList(\_dataContext.Brands, "Id", "Name" , product.BrandId);

return View(product);

}

}

phần biệt 2 action trùng tên trong 1 Class

[HttpGet] // dùng để di chuyển đến Controller /Action bằng đường dẫn

[HttpPost] // gửi dữ liệu từ form đến Controller /Action khi Submit trong thẻ form

* Khi nào dùng 2 action cùng tên trong Class : khi có thẻ form

A screenshot of a computer program

AI-generated content may be incorrect.

[HttpPost] // Chỉ nhận POST requests

* public IActionResult Create( Student item) // chỉ có Property của Student item trong function này sẽ ánh xạ đến

<**span** **asp-validation-for**="Name" class="text-danger"> </**span**>

* ModelState.IsValid **chỉ dựa vào dữ liệu từ form ban đầu**, không tự động thay đổi khi bạn gán object mới cho biến.

[HttpPost] // Chỉ nhận POST requests

public IActionResult Create( Student item) {

if (ModelState.IsValid)

{

return View("Index" , "Item");

}

return View (item);

}

View Create

@model Check.Models.Student

<**form** **asp-action**="Create" **asp-controller**="Item" method="post">

<**input** **asp-for**="Name" />

<**span** **asp-validation-for**="Name" class="text-danger"> </**span**>

<button type="submit">Lưu</button>

</**form**>

@section Scripts {

@{

await Html.PartialAsync("\_ValidationScriptsPartial");

}

}

Có hai cách thể hiện lỗi Property của Object để hiện lên <**span** **asp-validation-for**="Name" class="text-danger"> </**span**>

Cách 1 [Required, MinLength(4, ErrorMessage = "yêu cầu nhập tên Thương hiệu ")] //check điều kiện nếu không thỏa mãn điền kiện thì hiện message lỗi

Cách 2 ModelState.AddModelError("PropertyName", "stringContex");

Ví dụ cách 1

public class ProductModel

{

[Required(ErrorMessage = "Tên sản phẩm là bắt buộc")]

public string Name { get; set; }

[Range(1, 100000, ErrorMessage = "Giá phải lớn hơn 0")]

public decimal Price { get; set; }

}

<span asp-validation-for="Name" class="text-danger"></span>

<span asp-validation-for="Price" class="text-danger"></span>

→ Nó sẽ tự hiển thị lỗi.

Ví dụ cách 2

[HttpPost]

public IActionResult Create(ProductModel product)

{

if (product.Price <= 0)

{

ModelState.AddModelError("Price", "Giá phải lớn hơn 0");

}

if (product.Name == "Test")

{

ModelState.AddModelError("Name", "Tên 'Test' không được phép dùng");

}

if (!ModelState.IsValid)

{

return View(product);

}

// Lưu dữ liệu vào database

return RedirectToAction("Index");

}

Recap : truy cập bằng điều hường đến Controller /Action thì dùng [HttpGet] , truy cập điều hướng đến Controller /Action từ form thì dùng [HttpPost]