**#1\_ .NET Core vs .NET Framework**

|  |  |  |
| --- | --- | --- |
| **BASED ON** | **.NET core** | **.NET Framework** |
| **Open Source** | √ | × |
| **Cross-Platform** | Windows,Linux, Mac OS | Only windows |
| **Application Models** | Focus on Web,windows mobile, windows store | Desktop,web applications,windows forms, WPF |
| **Installation** | Packaged and installed independently,Cross-platform | A single package |
| **Support** | Microserives, Rest APIs | Rest APIs |
| **WCF Services** | × | √ |
| **Performance and Scalability** | high | less |
| **Android Development** | iOS, Android, Windows phones | × |
| **Packaging and Shipping** | .Net core is shipped as a collection of Nugget packages | All the libraries of .NET Framework are packaged and shipped together |
| **Deployment Model** | Updated instantly on one machine at at a time => flexible | Deployed on the Internet Information Server only |
| **Compatibility** | Windows,Linux, Mac OS | Only windows |
| **CLI Tools** |  |  |
| **Security** | × | √ |

**#2\_ Migration**

|  |  |
| --- | --- |
| **Package Manager Console** | |
| **Thêm migration** | Add-migration[name-migration] |
| **Cập nhật database từ migration** | Update-database  => Update-database -migration [migration-name] |
| **Xóa migration** | Remove-migration |

**#ASP.NET MVC**

**#3\_ Model Blinding (Ánh xạ dữ liệu) & Model Validation**

Controller và Razor pages làm việc với dữ liệu bằng HTTP requests, ví dụ, route data gửi record key, posts form fileds gửi giá trị **properties** model

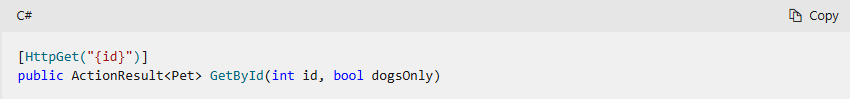
- Truy xuất dữ liệu (**Retrives data**) từ nhiều nguồn: route data, form flelds, and query strings.

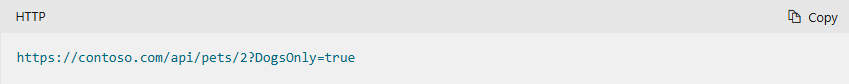
- Chuyển dữ liệu từ controllers và Razor pages vào các thông số (**parameters**) và **public properies**

- Chuyển đổi chuỗi dữ liệu sang kiểu .NET

- Cập nhật thuộc tính (**properties**) của các kiểu phức tạp (**complex types**)

Mục đích



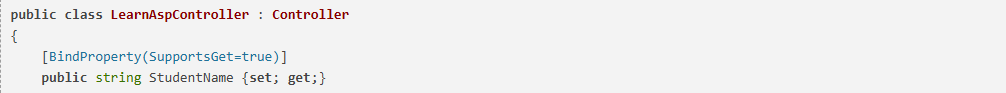


=> {“DogsOnly”= true; “id”=2 }

Khi hệ thống tìm từ nguồn đến, nếu không có sẽ thiết lập giá trị mặc định cho tham số của phương thức

Lỗi khi binding hoặc validation sẽ lưu trong **ControllerBase.ModelState** (dùng **ModelState.IsValid** để kiểm tra ánh xạ dữ liệu.

**[BindProperty] :** Để nguồn dữ liệu qua **HTTP requests** (Form, route data, url request,…)tự động convert dữ liệu và gán vào thuộc tính của Controller



**- [FormQuery] :** dữ liệu trích xuất từ Url query (/abc/?key=value)

**- [FormRoute] :** dữ liệu lấy từ giá trị trong tham số của Route

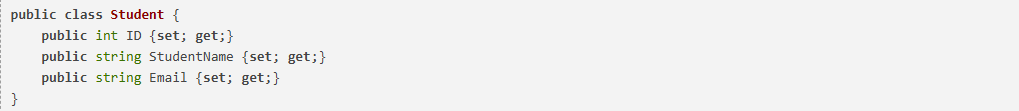
**- [FormForm] :** dữ liệu lấy từ Form

**- [FormBody]:** dữ liệu lấy từ Body của Http Request

**- [FormHeader] :** dữ liệu lấy từ Header của Http Request

**Name trong [BindingProperty], [FromQuery], [FromRoute] … :** khi binding dữ liệu mà tên ở nguồn đến và tên thuộc tính Controller khác nhau dùng thuộc tính Name







=> prefix.property\_name (tên key được tổ hợp từ tiền tố thuộc tính và tên thuộc tính).