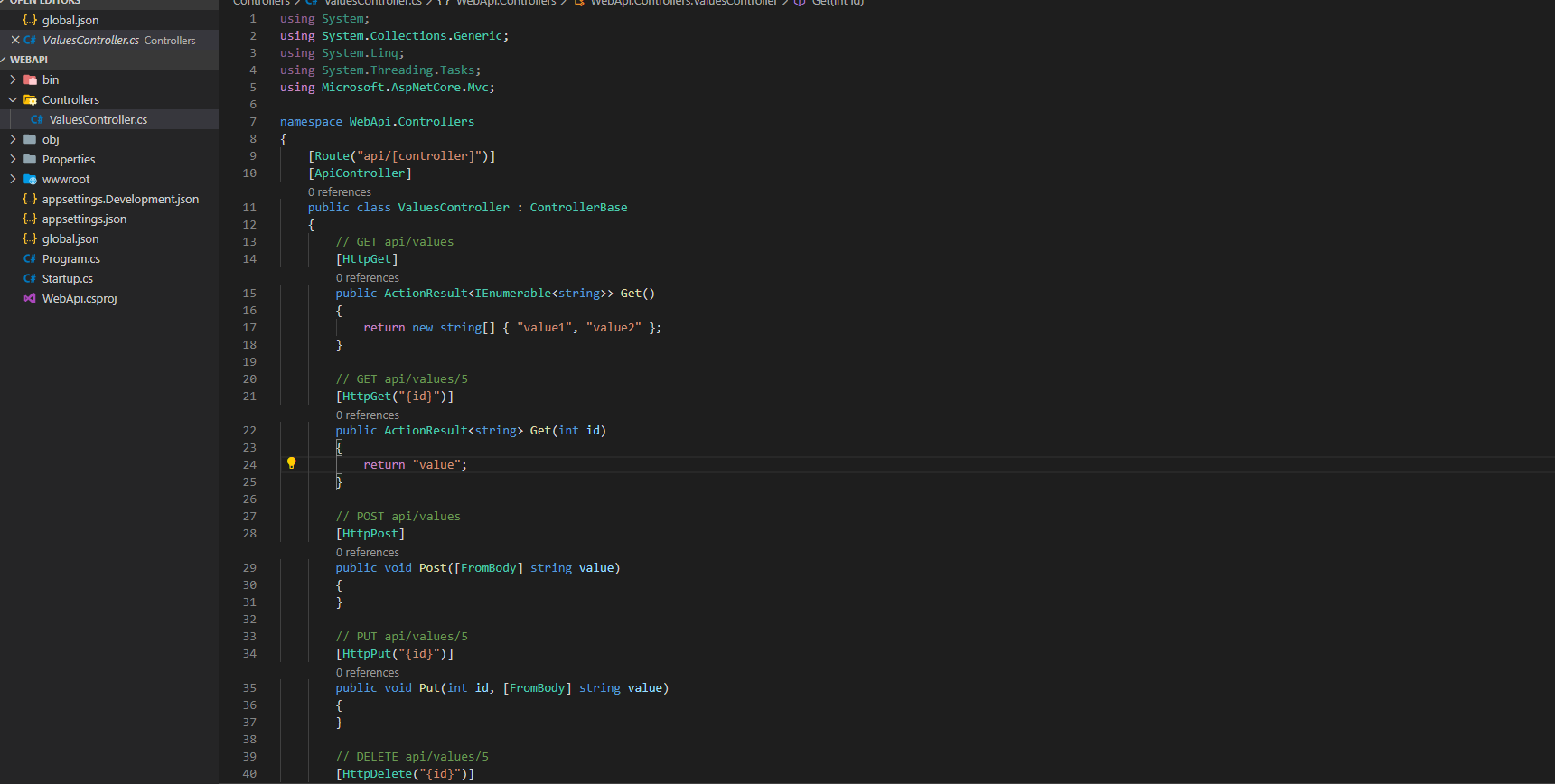
<https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-3.1>

Controller nằm giữa route và view làm nhiệm vụ xử lý ở giữa

Controller nằm trong thư mục Controllers, tên của Controller có hậu tố Controller và thừa kế từ ControlerBase



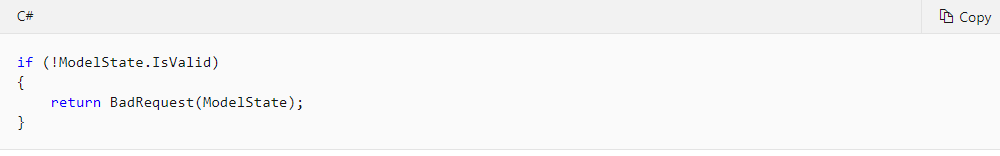


Dùng attribute chỉ ra

Route api/[controler] phần [controler] được quết định bởi các method, cái này được bật từ [ApiController]

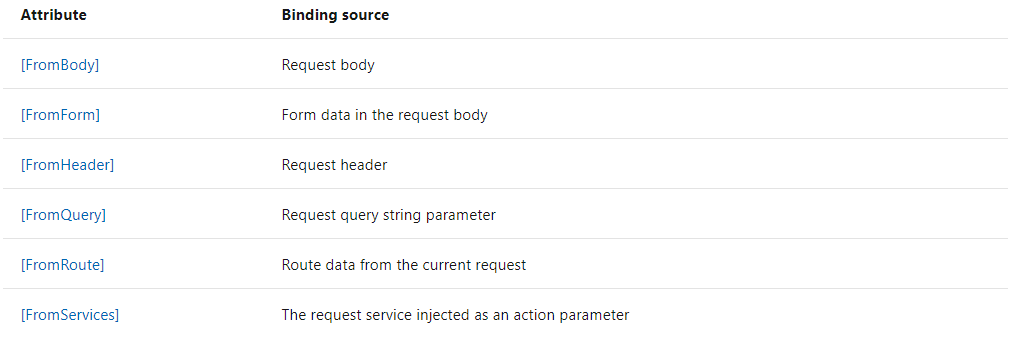
[ApiController] : đưa vào để apply các chức năng của Api Controller : routing, 400 response, param inference, Multipart/form-data request inference, error detail

Class controller kết thúc bằng hậu tố Controller và inherit từ ControllerBase

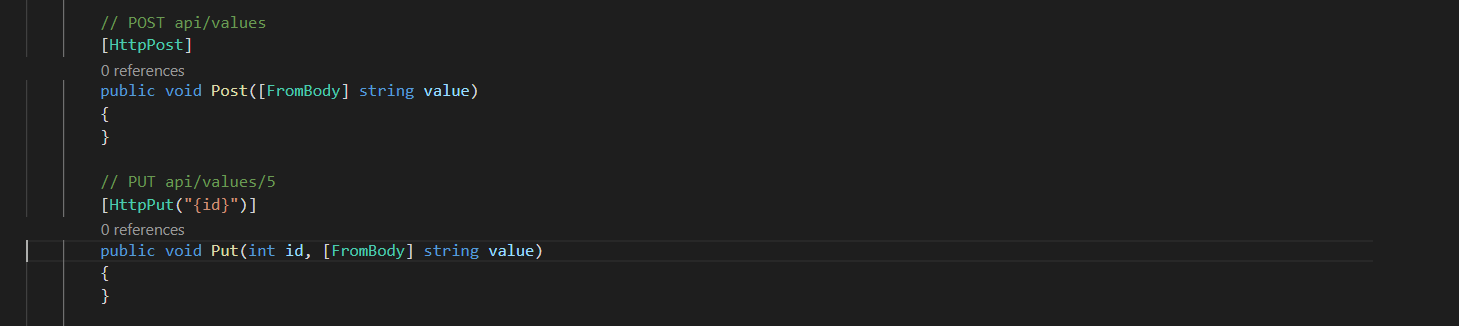


HTTP 400 responses: sẽ được tự động trả về, vì thế không cần code như trên để handler trường hợp này

Binding source parameter inference: đính kèm các ràng buộc về param



Định nghĩa vị trí để lấy param

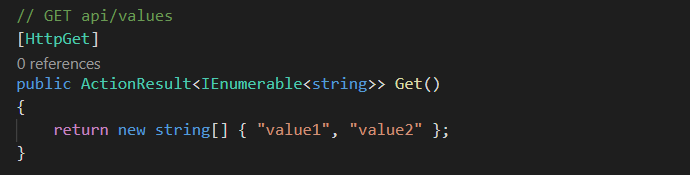


Những biến được lấy từ các vị trị khác url, thì sẽ được định nghĩa thêm attribute này

[FromForm] attribute sẽ chỉ ra dữ liệu được lấy theo multipart/form-data

Define supported request content types with the [Consumes] attribute: dùng attribute Consumes để định nghĩa kiểu dữ liệu sử dụng, json, xml, x-www-form-urlencoded

Các method căn bản:



[HttpGet] chỉ ra action get

IEnumerable<T>: dữ liệu trả về ở đây là 1 cái array string, nếu là string thì ko cần Ienumerable. IEnumerable<T>: nghĩa là return về cái gì đó với các phần tử string ví dụ array, list, tupple,

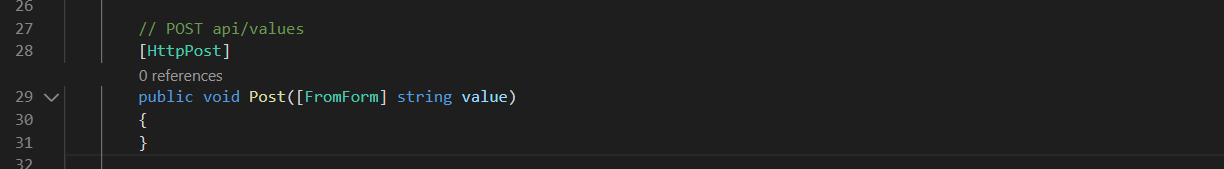
ActionResult<T>: giá trị trả về có thể là specific type( kết quả) ActionResult( trạng thái action)



Truyền vào biến id trong method lấy biến này để sử dụng

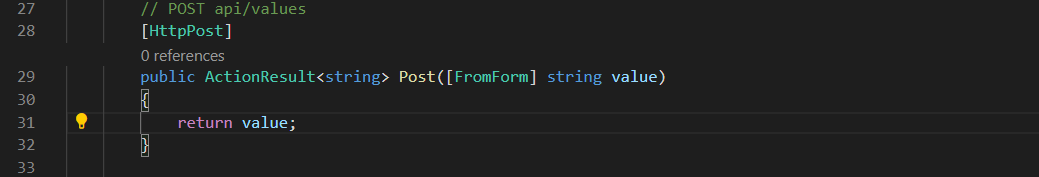
Giá trị trả về chỉ là string

Và có dùng ActionResult

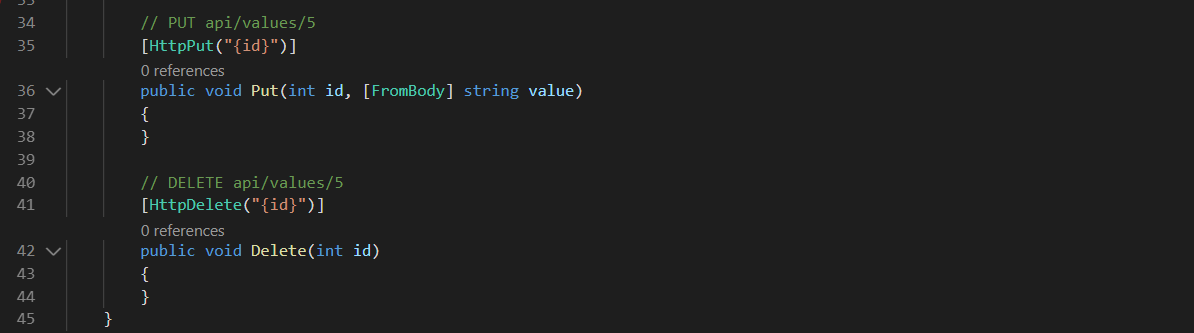


Post sử dùng thêm attribute để biding source parameter inference

ở đây post không trả về nên trả về là void



Khi có trả về



Tương tự với put và delete

ở đây các link có thể trùng nhau, nhưng dựa vào method để nhận ra link đó