

- BindRequired e BindNever

Estes atributos permitem definir se um valor vai ser vinculado ou não

O atributo *BindRequired* diz ao model binder que a vinculação do atributo é obrigatório e se isso não ocorrer vai gerar um erro.

O atributo *BindNever* diz ao model binder que não deve vincular o atributo especificado.

1- BindRequired

```
[HttpGet("{id:int:min(1)}", Name = "ObterProduto")]
public async Task<ActionResult<Produto>> Get(int id, [BindRequired] string nome)
{
    var nomeProduto = nome;

    var produto = await _context.Produtos.FirstOrDefaultAsync(p => p.ProdutoId == id);

    if (produto is null)
    {
        return NotFound("Produto não encontrado...");
    }
    return produto;
}
```

2- BindNever

```
public class Categoria
{
    public Categoria()
    {
        Produtos = new Collection<Produto>();
    }

    public int CategoriaId { get; set; }
    public string? Nome { get; set; }

    [BindNever]
    public string? ImagemUrl { get; set; }

    public ICollection<Produto>? Produtos { get; set; }
}
```

Usando os atributos

1- FromForm

Este atributo faz com que o Model Binder utilize somente os dados recebidos do formulário enviado.

```
public IActionResult Detail([FromForm] ProdutoViewModel produtoViewModel) =>
    View(produtoViewModel);
```

2- FromRoute

Este atributo faz com que o Model Binder apenas vincule dados que são oriundos da rota de dados

```
public IActionResult Detail([FromRoute] int id) => View();
```

3- FromQuery

Este atributo diz ao Model Binder para receber apenas os dados da cadeia de consulta (querystring).

```
public IActionResult Detail([FromQuery] int id) => View();
```

Se fizermos uma solicitação com a URL: Produto/Detalhe/2?Id=4

O valor 4 será vinculado e não o valor 2 conforme seria previsto no comportamento padrão.

4- FromHeader

Este atributo diz ao Model Binder para vincular os valores que vêm no cabeçalho da requisição HTTP.

5- FromBody

Este atributo diz ao Model Binder para vincular dados a partir do Body do request.

```
public IActionResult Detail([FromBody] ProdutoViewModel produtoViewModel) => View(produtoViewModel);
```

6- FromServices

Este atributo vincula o valor especificado à implementação que foi configurada no seu container de injeção de dependência.

```
public IActionResult Detail([FromServices] IPrintable printer) => View();
```

Exemplo de atributo [FromQuery] em ProdutosController

```
[HttpGet("{id}", Name = "ObterProduto")]
public async Task<ActionResult<Produto>> Get([FromQuery] int id)
{
    var produto = await _context.Produtos.FirstOrDefaultAsync(p => p.ProdutoId == id);

    if (produto is null)
    {
        return NotFound("Produto não encontrado...");
    }
    return produto;
}
```

Para obter o valor do id agora devemos informar na url a querystring: ?id=2

Ex: <https://localhost:7278/produtos/1?id=2>