* **[HttpDelete]**
* **[HttpGet]**
* **[HttpHead]**
* **[HttpOptions]**
* **[HttpPatch]**
* **[HttpPost]**
* **[HttpPut]**

[RoutePrefix("orders")]

public class OrdersController : ApiController

{

[Route("{id:int}")] // constrained parameter

public HttpResponseMessage Get(int id) { ... }

[Route("details")] // literal

public HttpResponseMessage GetDetails() { ... }

[Route("pending", RouteOrder = 1)]

public HttpResponseMessage GetPending() { ... }

[Route("{customerName}")] // unconstrained parameter

public HttpResponseMessage GetByCustomer(string customerName) { ... }

[Route("{\*date:datetime}")] // wildcard

public HttpResponseMessage Get(DateTime date) { ... }

}

These routes are ordered as follows.

1. orders/details
2. orders/{id}
3. orders/{customerName}
4. orders/{\*date}
5. orders/pending

[Route("api/books")]

public IEnumerable<Book> GetBooks() { ... }

[Route("api/books/{id:int}")]

public Book GetBook(int id) { ... }

[Route("api/books")]

[HttpPost]

public HttpResponseMessage CreateBook(Book book) { ... }

//GET method

//single parameter

public IActionResult Get(int id)

//multiple parameter

[HttpGet("{id}/{backendOnly}")]

public IActionResult Get(int id, string backendOnly)

//multiple parameters

[HttpGet("{id}/{first}/{second}")]

public IActionResult Get(int id, string first, string second)

C#

//GET request using route data

<https://localhost:44363/2>

<https://localhost:44363/2/first>

<https://localhost:44363/2/first/second>

Passing Multiple Parameters Using FromQuery

In this option the parameters are bound using the request query string.  
  
*Example:*

//GET method

//single parameter

[HttpGet("details")]

public IActionResult Details(int id)

//multiple parameter

[HttpGet("details")]

public IActionResult Details(int id, string first)

//multiple parameters

[HttpGet("details")]

public IActionResult Details(int id, string first, string second)

C#

//GET request using query parameters

<https://localhost:44363/details?id=2>

<https://localhost:44363/details?id=2&&first=csharp>

<https://localhost:44363/details?id=2&&first=csharp&&second=mvc>

What is Model Binding?

The *model binding* works with data coming from HTTP requests and passed to the controller and Razor pages as parameters. It works in the following ways:

* It fetches data from various sources like query string, form fields and data routes
* It provides data to controller methods as parameter
* It converts data to .NET types
* It updates properties of complex types

Model Binding Sources

It gets the data from various sources in the form of key-value pairs from the following sources:

* Form fields
* Request body
* Route data parameters
* Query string parameters
* Uploaded files

To work with *model binding*, we need to use following attributes:

* **BindProperties** attribute—This can be applied to class le