C# 將多個引數傳遞給 get 方法

[C#](https://www.delftstack.com/zh-tw/tags/c/) [C# Method](https://www.delftstack.com/zh-tw/tags/c#-method/)

創建時間: June-09, 2020 | 更新時間: June-25, 2020

1. [在 C# 中使用控制器動作將多個引數傳遞給 Get 方法](https://www.delftstack.com/zh-tw/howto/csharp/how-to-pass-multiple-parameters-to-a-get-method-in-csharp/#%25E5%259C%25A8-c%23-%25E4%25B8%25AD%25E4%25BD%25BF%25E7%2594%25A8%25E6%258E%25A7%25E5%2588%25B6%25E5%2599%25A8%25E5%258B%2595%25E4%25BD%259C%25E5%25B0%2587%25E5%25A4%259A%25E5%2580%258B%25E5%25BC%2595%25E6%2595%25B8%25E5%2582%25B3%25E9%2581%259E%25E7%25B5%25A6-get-%25E6%2596%25B9%25E6%25B3%2595)
2. [使用屬性路由將多個引數傳遞給獲取方法的程式碼](https://www.delftstack.com/zh-tw/howto/csharp/how-to-pass-multiple-parameters-to-a-get-method-in-csharp/#%25E4%25BD%25BF%25E7%2594%25A8%25E5%25B1%25AC%25E6%2580%25A7%25E8%25B7%25AF%25E7%2594%25B1%25E5%25B0%2587%25E5%25A4%259A%25E5%2580%258B%25E5%25BC%2595%25E6%2595%25B8%25E5%2582%25B3%25E9%2581%259E%25E7%25B5%25A6%25E7%258D%25B2%25E5%258F%2596%25E6%2596%25B9%25E6%25B3%2595%25E7%259A%2584%25E7%25A8%258B%25E5%25BC%258F%25E7%25A2%25BC)
3. [使用 [FromQuery]將程式碼段傳遞給 Get 方法的程式碼](https://www.delftstack.com/zh-tw/howto/csharp/how-to-pass-multiple-parameters-to-a-get-method-in-csharp/#%25E4%25BD%25BF%25E7%2594%25A8-fromquery%25E5%25B0%2587%25E7%25A8%258B%25E5%25BC%258F%25E7%25A2%25BC%25E6%25AE%25B5%25E5%2582%25B3%25E9%2581%259E%25E7%25B5%25A6-get-%25E6%2596%25B9%25E6%25B3%2595%25E7%259A%2584%25E7%25A8%258B%25E5%25BC%258F%25E7%25A2%25BC)

ASP.NET MVC 用於開發 Web 應用程式。MVC Web API 和網頁框架現已合併為一個名為 MVC 6 的框架。MVC 是一種模式，允許你通過路由的請求與模型，檢視和控制器進行互動。

在本文中，我們將討論用於將多個引數傳遞給 MVC 控制器中的 Get 方法的各種方法。

[在 C# 中使用控制器動作將多個引數傳遞給 Get 方法](https://www.delftstack.com/zh-tw/howto/csharp/how-to-pass-multiple-parameters-to-a-get-method-in-csharp/#%E5%9C%A8-c-%E4%B8%AD%E4%BD%BF%E7%94%A8%E6%8E%A7%E5%88%B6%E5%99%A8%E5%8B%95%E4%BD%9C%E5%B0%87%E5%A4%9A%E5%80%8B%E5%BC%95%E6%95%B8%E5%82%B3%E9%81%9E%E7%B5%A6-get-%E6%96%B9%E6%B3%95)

控制器動作是一種用於處理傳入請求的方法。然後返回動作結果。動作結果是對傳入請求的響應。在這種情況下，我們使用控制器動作將多個引數傳遞給 Get 方法。

示例程式碼：

public string Get(int? id, string FirstName, string LastName, string Address)

{

if (id.HasValue)

GetById(id);

else if (string.IsNullOrEmpty(Address))

GetByName(FirstName, LastName);

else

GetByNameAddress(FirstName, LastName, Address);

}

[使用屬性路由將多個引數傳遞給獲取方法的程式碼](https://www.delftstack.com/zh-tw/howto/csharp/how-to-pass-multiple-parameters-to-a-get-method-in-csharp/#%E4%BD%BF%E7%94%A8%E5%B1%AC%E6%80%A7%E8%B7%AF%E7%94%B1%E5%B0%87%E5%A4%9A%E5%80%8B%E5%BC%95%E6%95%B8%E5%82%B3%E9%81%9E%E7%B5%A6%E7%8D%B2%E5%8F%96%E6%96%B9%E6%B3%95%E7%9A%84%E7%A8%8B%E5%BC%8F%E7%A2%BC)

在屬性路由中，我們使用屬性來定義路由。這是推薦的方法，因為它減少了出錯的機會。它使你可以更好地控制 Web API 中的 URL。

使用此方法的正確語法如下：

//Specify route

[Route("api/YOURCONTROLLER/{parameterOne}/{parameterTwo}")]

public string Get(int parameterOne, int parameterTwo)

{

return $"{parameterOne}:{parameterTwo}";}

}

示例程式碼：

//Specify route

[Route("api/yourControllerName/{FirstName}/{LastName}/{Address}")]

public string Get(string id, string FirstName, string LastName, string Address)

{

return $"{FirstName}:{LastName}:{Address}";

}

[使用 [FromQuery]將程式碼段傳遞給 Get 方法的程式碼](https://www.delftstack.com/zh-tw/howto/csharp/how-to-pass-multiple-parameters-to-a-get-method-in-csharp/#%E4%BD%BF%E7%94%A8-fromquery-%E5%B0%87%E7%A8%8B%E5%BC%8F%E7%A2%BC%E6%AE%B5%E5%82%B3%E9%81%9E%E7%B5%A6-get-%E6%96%B9%E6%B3%95%E7%9A%84%E7%A8%8B%E5%BC%8F%E7%A2%BC)

[FromQuery] 用於指定引數應與請求查詢字串繫結。它是 FromQueryAttribute 類的屬性。

使用此屬性的正確語法如下：

[HttpGet]

public string GetByAttribute([FromQuery]string parameterOne, [FromQuery]string parameterTwo)

{

}

示例程式碼：

[Route("api/person")]

public class PersonController : Controller

{

[HttpGet]

//Passing one parameter to the Get method

public string GetById([FromQuery]int id)

{

}

[HttpGet]

//Passing two parameters to the Get method

public string GetByName([FromQuery]string firstName, [FromQuery]string lastName)

{

}

[HttpGet]

//Passing multiple parameters to the Get method

public string GetByNameAndAddress([FromQuery]string firstName, [FromQuery]string lastName, [FromQuery]string address)

{

}

}