Read URL parameters in a C# application

ASP.NET Core MVC/Web API

In an ASP.NET Core MVC or Web API application, you can retrieve URL parameters from route data or query string parameters.

For Route Parameters:

```
// in a controller action
public IActionResult Index(int id_param)
{
  // 'id_param' is a route parameter
  var theParam = id_param.ToString();
  return View();
}
```

For Query String Parameters:

```
// in a controller action
public IActionResult Index()
{
  var queryParam = Request.Query["id_param"].ToString();
  return View();
}
```

ASP.NET (web form)

In an ASP.NET web form application, you can access query string parameters from the Request object in the page code behind.

```
// in your page code-behind
protected void Page_Load(object sender, EventArgs e)
{
  string paramValue = Request.QueryString["id_param"];
}
```

Console Application

If you are processing URL parameters in a console application, you can handle them as part of the input string.

```
// example of parsing URL parameters from a string
using System;
using System.Web;

class Program
{
  static void Main()
  {
    string url = "https://example.com?page=1&sort=asc";
    var uri = new Uri(url);
    var query = HttpUtility.ParseQueryString(uri.Query);

    string page = query["page"];
    string sort = query["sort"];
    Console.WriteLine($"Page: {page}, Sort: {sort}");
  }
}
```

ASP.NET Core Minimal API

If using the Minimal API approach in ASP.NET Core you can look at route parameters.

```
var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/items/{id_param}", (int id_param) =>
{
  // 'id' is a route parameter
  return Results.Ok($"Item ID: {id_param}");
});

app.MapGet("/search", (HttpContext context) =>
{
  var query = context.Request.Query;
  string searchTerm = query["q"];
  return Results.Ok($"Search term: {searchTerm}");
});

app.Run();
```