# How to edit QueryString in ASP.NET Core

## Introduction

This sample demonstrates how to edit QueryString in ASP.NET Core.

## Sample prerequisites

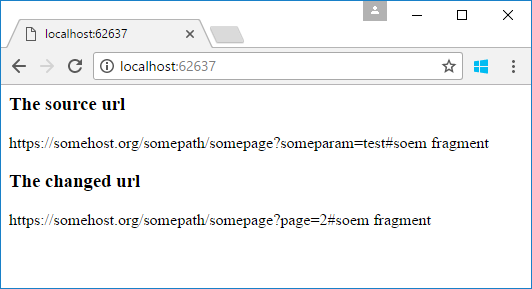
* Visual Studio 2017 or above. [[Visual Studio Home Page](https://www.visualstudio.com/)]
* Visual Studio enabled ASP.NET Core develop component.

## Building the sample

Use Visual Studio to open the sample solution **CSEditQueryStringAspNetCore**, then press F6 Key to build the sample project.

## Running the sample

* Use Visual Studio to open the sample solution **CSEditQueryStringAspNetCore**, then press F5 Key or select Debugging from the menu.
* When the project is running, you will see below page in browser.



## Using the code

QueryParameterEditor.cs

public class QueryParameterEditor

{

private List<QueryParameter> \_querys = new List<QueryParameter>();

public string Authority { get; set; }

public string QueryString

{

get

{

return string.Join("&", \_querys);

}

}

public string Fragment { get; set; }

public string[] AllKeys

{

get

{

return \_querys.Select(m => m.Key).ToArray();

}

}

public string this[string key]

{

get

{

var param = \_querys.FirstOrDefault(m => m.Key.Equals(key, StringComparison.CurrentCultureIgnoreCase));

return param == null ? null : param.Value;

}

set { SetQueryParam(key, value); }

}

public QueryParameterEditor(string uriString)

{

string authority, queryString, fragment;

uriString.CutStringByFirstKey("?", out authority, out queryString);

queryString.CutStringByFirstKey("#", out queryString, out fragment);

Authority = authority;

InitParameters(queryString);

Fragment = fragment;

}

public QueryParameterEditor SetQueryParam(string key, string value)

{

var param = new QueryParameter { Key = key, Value = value };

SetQueryParam(param);

return this;

}

public QueryParameterEditor SetQueryParam(params QueryParameter[] parameters)

{

foreach (var item in parameters)

{

var queryParam = \_querys.FirstOrDefault(m => m.Key.Equals(item.Key, StringComparison.CurrentCultureIgnoreCase));

if (queryParam == null)

{

\_querys.Add(item);

}

else

{

queryParam.Value = item.Value;

}

}

return this;

}

public QueryParameterEditor RemoveQueryParam(params string[] keys)

{

foreach (var key in keys)

{

var queryParam = \_querys.FirstOrDefault(m => m.Key == key);

\_querys.Remove(queryParam);

}

return this;

}

public override string ToString()

{

return Authority +

(string.IsNullOrEmpty(QueryString) ? "" : $"?{QueryString}") +

(string.IsNullOrEmpty(Fragment) ? "" : $"#{Fragment}");

}

private void InitParameters(string queryString)

{

string[] queryArr = queryString.Split('&');

foreach (var item in queryArr)

{

if (!string.IsNullOrEmpty(item))

{

string key, value;

item.CutStringByFirstKey("=", out key, out value);

\_querys.Add(new QueryParameter { Key = key, Value = value });

}

}

}

}

public class QueryParameter

{

public string Key { get; set; }

public string Value { get; set; }

public override string ToString()

{

return $"{Key}={Value}";

}

}

HomeController.cs

public IActionResult Index()

{

string originalUrl = "https://somehost.org/somepath/somepage?someparam=test#soem fragment";

var qpe = new QueryParameterEditor(originalUrl);

qpe["page"] = "1";

qpe.SetQueryParam("page", "2").RemoveQueryParam("someparam");

ViewBag.OriginalUrl = originalUrl;

ViewBag.EditedUrl = qpe.ToString();

return View();

}

Index.cshtml

<h3>The source url</h3>

<p>@ViewBag.OriginalUrl</p>

<h3>The changed url</h3>

<p>@ViewBag.EditedUrl</p>