

# WebAPI2CLI

# Contents

<b>WebAPI2CLI</b>	<b>1</b>
<b>Why</b>	<b>2</b>
<b>How to use ( for .NET Core 3.1 )</b>	<b>2</b>
Step 0 : install into your ASP.NET Core Web . . . . .	2
Step 1 - find and save the definition of the commands, i.e. WebAPI endpoints . . . . .	2
Step 2 - run the commands . . . . .	3
<b>If you want a demo</b>	<b>3</b>
If you have Win10 x64 . . . . .	3
If you have another operating system than Win10-x64 and want a demo	3
Could I see an example of the output ? . . . . .	4
Could you put here the output ? . . . . .	4
<b>F.A.Q.</b>	<b>5</b>
I want to contribute . Where is the code ? . . . . .	5
Will my WebAPI work as before? . . . . .	5
I want to run as a CLI and then use the WebAPI as before. Could I do that? . . . . .	5
I found a bug / I need a feature . Where can I report ? . . . . .	5
Where can I download this document ? . . . . .	5

# WebAPI2CLI

## Execute ASP.NET Core WebAPI from Command Line



## Why

What if, instead of running the WebAPI ( or just the site ) and waiting for commands from the user, you want also to execute from the command line some controllers actions ?

This project let's you do that by enabling the command line with

```
< myexe >.exe --CLI_ENABLED=1 --CLI_Commands=" ... "
```

The command names are in a *cli.txt* file that can be generated with

```
< myexe >.exe --CLI_ENABLED=1 --CLI_HELP=1
```

## How to use ( for .NET Core 3.1 )

### Step 0 : install into your ASP.NET Core Web

Install the package <https://www.nuget.org/packages/ExtensionNetCore3>

Modify your ASP.NET Core as below:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddCLI();
    //your code omitted
}
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    app.UseCLI();
    //your code omitted
}
```

And that is all modifications that you need to do for the source code.

### Step 1 - find and save the definition of the commands, i.e. WebAPI endpoints

First, you must generate the definition of the commands. For this, we take the OPEN API (swagger ) approach.

For this, after you compile the project, you will run your .exe program with arguments:

```
< myexe >.exe --CLI_ENABLED=1 --CLI_HELP=1
```

( or make this from *Visual Studio, Project, Properties, Debug* )

This will generate a *cli.txt* file with all definitions of the WebAPI. ( if your API does not appear, check if you have *ApiController* defined)

Open your *cli.txt* file and modify the names of the commands as you wish (also , the arguments )

Copy this *cli.txt* in your solution and be sure that is copied with the exe ( in Visual Studio right click the file, properties, Build Action = Content, CopyToOutputDirectory = Copy if newer)

## Step 2 - run the commands

Ensure that the file is near your exe WebAPI.

Run the exe with the following:

```
< myexe >.exe --CLI_ENABLED=1 --CLI_Commands="your first command,your second command, and enumerate all commands"
```

The program will run the commands and output the result.

More details here( including a demo)

<https://ignatandrei.github.io/WebAPI2CLI/>

## If you want a demo

### If you have Win10 x64

Go to [https://dev.azure.com/ignatandrei0674/WebAPI2CLI/\\_build?definitionId=7&\\_a=summary](https://dev.azure.com/ignatandrei0674/WebAPI2CLI/_build?definitionId=7&_a=summary)

Click on the latest job.

Click on artifacts.

See drop1 . In the right of drop1 , you can download the drop1 folder.

Unzip the drop1 ( you will need the cli.txt that contains the WebAPI definitions)

Inside drop1 , you will find TestWebAPISite.exe

Run TestWebAPISite.exe and browse to <http://localhost:5000/swagger>

Close TestWebAPISite.exe command prompt.

Now run

```
TestWebAPISite.exe --CLI_Enabled=1 --CLI_COMMANDS="GetMathId_Http,WeatherGet"
```

( ensure the cli.txt is near to TestWebAPISite.exe)

### If you have another operating system than Win10-x64 and want a demo

It is easy to make a demo for you. Modify the .csproj and the yml file in azure. Or make an issue at <https://github.com/ignatandrei/webAPI2CLI/issues> and I will do it for you.

## Could I see an example of the output ?

Of course. Every build in AzureDevOps has a last step , runningADemo. See [https://dev.azure.com/ignatandrei0674/WebAPI2CLI/\\_build?definitionId=7&\\_a=summary](https://dev.azure.com/ignatandrei0674/WebAPI2CLI/_build?definitionId=7&_a=summary)

## Could you put here the output ?

Yes. This is the output of

TestWebAPISite.exe --CLI\_ENABLED=1 --CLI\_Commands="GetMathId\_Http,MathPOST"

See *Result* variable

1. 2020-03-10T19:03:39.2929915Z ExtensionNetCore3 version:1.2020.10310.11900
2. 2020-03-10T19:03:40.0045036Z info: Microsoft.Hosting.Lifetime[0]
3. 2020-03-10T19:03:40.0046360Z Now listening on: http://localhost:5000
4. 2020-03-10T19:03:40.0047178Z info: Microsoft.Hosting.Lifetime[0]
5. 2020-03-10T19:03:40.0047815Z Now listening on: https://localhost:5001
6. 2020-03-10T19:03:40.0048438Z info: Microsoft.Hosting.Lifetime[0]
7. 2020-03-10T19:03:40.0049078Z Application started. Press Ctrl+C to shut down.
8. 2020-03-10T19:03:40.0049702Z info: Microsoft.Hosting.Lifetime[0]
9. 2020-03-10T19:03:40.0050292Z Hosting environment: Production
10. 2020-03-10T19:03:40.0050877Z info: Microsoft.Hosting.Lifetime[0]
11. 2020-03-10T19:03:40.0051895Z Content root path: D:\a\1\
12. 2020-03-10T19:03:44.6267548Z CLIExecute version:1.2020.10310.11900
13. 2020-03-10T19:03:45.4122334Z *executing GetMathId\_Http*
14. 2020-03-10T19:03:46.0585725Z {
15. 2020-03-10T19:03:46.0586788Z "Command": {
16. 2020-03-10T19:03:46.0587511Z "NameCommand": "GetMathId\_Http",
17. 2020-03-10T19:03:46.0588202Z "Host": "http://localhost:5000",
18. 2020-03-10T19:03:46.0588851Z "RelativeRequestUrl": "api/MathAdd/5",
19. 2020-03-10T19:03:46.0589427Z "Verb": "GET"
20. 2020-03-10T19:03:46.0589936Z },
21. 2020-03-10T19:03:46.0590434Z "StatusCode": 200,
22. 2020-03-10T19:03:46.0590955Z "*Result*": "value5"
23. 2020-03-10T19:03:46.0591452Z }
24. 2020-03-10T19:03:46.0591977Z *executing MathPOST*
25. 2020-03-10T19:03:46.1304847Z {
26. 2020-03-10T19:03:46.1306343Z "Command": {
27. 2020-03-10T19:03:46.1307062Z "NameCommand": "MathPOST",
28. 2020-03-10T19:03:46.1307766Z "Host": "http://localhost:5000",
29. 2020-03-10T19:03:46.1308471Z "RelativeRequestUrl": "api/MathAdd",
30. 2020-03-10T19:03:46.1309710Z "Verb": "POST"
31. 2020-03-10T19:03:46.1310813Z },
32. 2020-03-10T19:03:46.1311457Z "StatusCode": 200,
33. 2020-03-10T19:03:46.1312079Z "Result": ""
34. 2020-03-10T19:03:46.1312652Z }

## **F.A.Q.**

### **I want to contribute . Where is the code ?**

All code source is at <https://github.com/ignatandrei/WebAPI2CLI/>

Please see issues tab if you want to know what needs development .

### **Will my WebAPI work as before?**

The software takes care about

`--CLI_ENABLED=1`

If you do not have this command, your website runs as before

### **I want to run as a CLI and then use the WebAPI as before. Could I do that?**

Yes. Use

```
< myexe >.exe --CLI_ENABLED=1 --CLI_Commands="your first com-  
mand,your second command" --CLI_STAY=1
```

### **I found a bug / I need a feature . Where can I report ?**

You can report problems at <https://github.com/ignatandrei/WebAPI2CLI/issues>

### **Where can I download this document ?**

There is a PDF at <https://ignatandrei.github.io/WebAPI2CLI/Web2CLI.pdf>