

To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.

Understanding launchSettings.json File Configuration



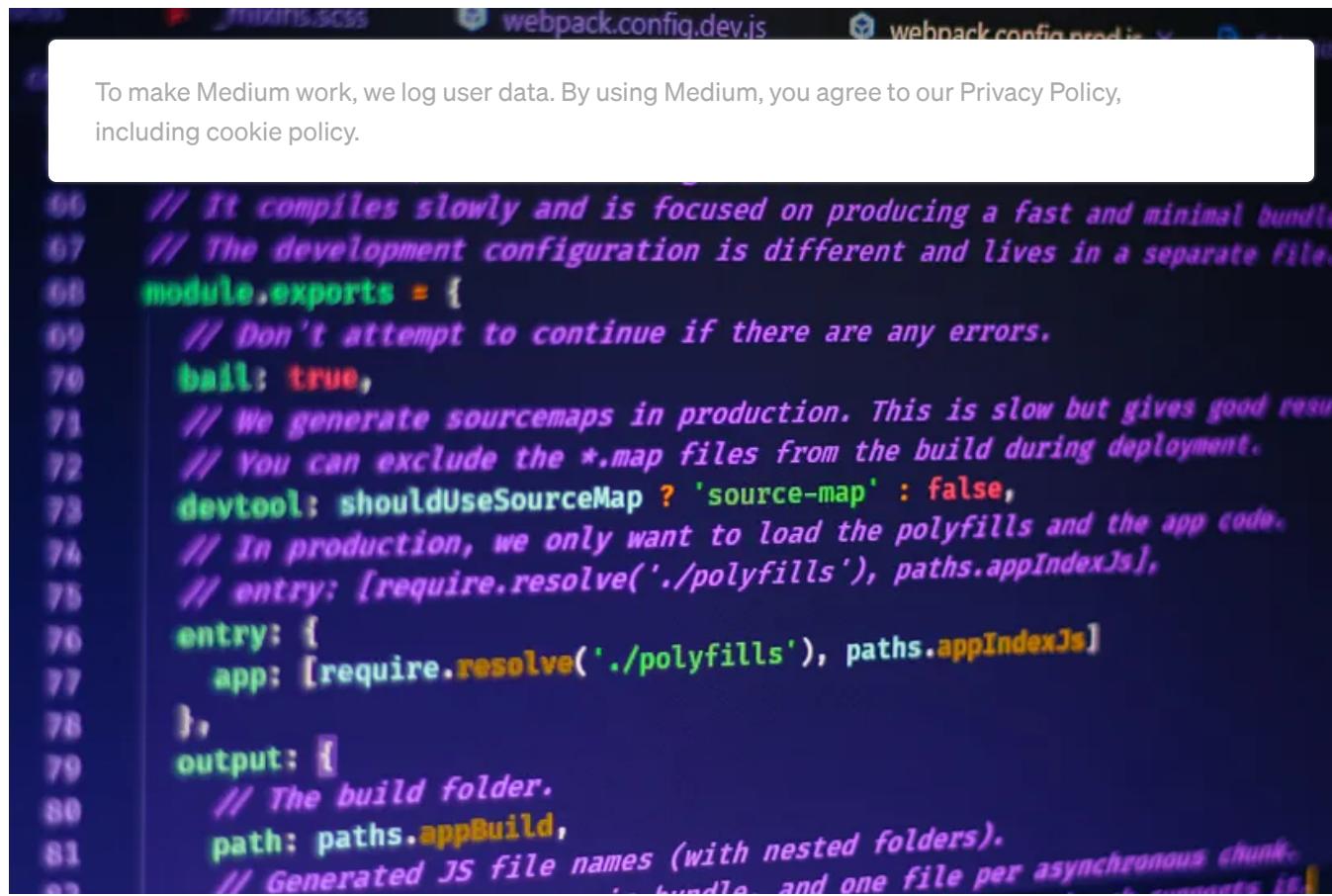
Jin Vincent Necessario · [Follow](#)

3 min read · Oct 30, 2022



340





```
66 // It compiles slowly and is focused on producing a fast and minimal bundle.
67 // The development configuration is different and lives in a separate file.
68 module.exports = {
69   // Don't attempt to continue if there are any errors.
70   bail: true,
71   // We generate sourcemaps in production. This is slow but gives good results.
72   // You can exclude the *.map files from the build during deployment.
73   devtool: shouldUseSourceMap ? 'source-map' : false,
74   // In production, we only want to load the polyfills and the app code.
75   // entry: [require.resolve('./polyfills'), paths.appIndexJs],
76   entry: {
77     app: [require.resolve('./polyfills'), paths.appIndexJs]
78   },
79   output: {
80     // The build folder.
81     path: paths.appBuild,
82     // Generated JS file names (with nested folders).
83     // i.e.: bundle.js, main.js, vendor.js, vendor-common.js, etc.
84     // One file per chunk.
85     filename: '[name].js',
86     chunkFilename: '[name].js',
87     // One file per asynchronous chunk.
88    异步文件名: '[name].js',
89    异步chunk文件名: '[name].js',
90     // One file per comments block.
91     comments: true
92   }
}
```

Photo by [Ferenc Almasi](#) on [Unsplash](#)

This article assumes that you have already created an ASP.NET or Web API Core project. If you haven't done it yet, I suggest you create one or open an existing ASP.NET Core project.

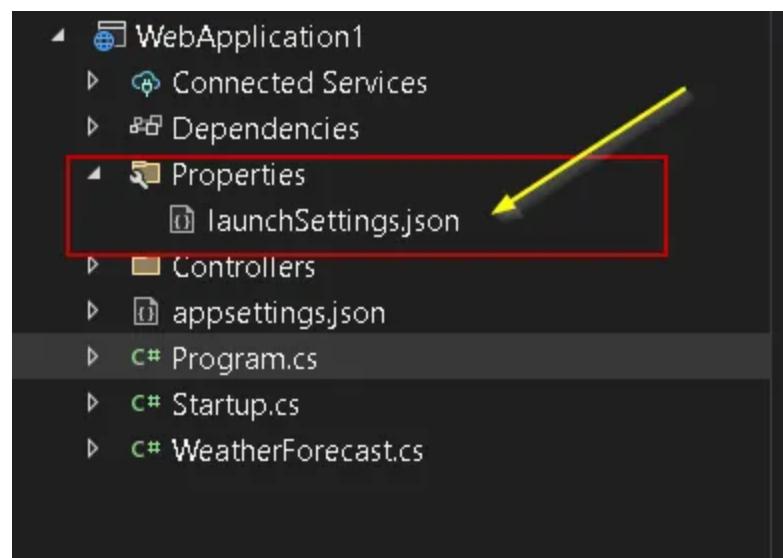
Every time we have created a with ASP.NET Core or Web API Core project the *launchSettings.json* file is also part of the creation process and it can be found

in the Properties section of the Solution Explorer window

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

If:

you're maybe curious about its purpose. You came to the right place, we'll try to investigate this file.



launchSetting.json file

Understanding `launchSettings.json` File

If we're going to open this configuration file (`launchSettings.json`), you'll be able to determine that it helps Visual Studio how to behave when launching an ASP.NET Core application.

Here are some points to keep in mind about this file

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

- In other words, this file isn't used or required when your project is published to the production server.
- The settings within the file are important for Visual Studio and if run the application using the .NET Core CLI. When using the .NET Core CLI see the command below.

```
dotnet run
```

Let's try to see an example of it below.

```
{
  "$schema": "http://json.schemastore.org/launchsettings.json",
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:57005",
      "sslPort": 44385
    }
  },
  "profiles": {
    "IIS Express": {
```

```
"commandName": "IISExpress",
```

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

```
    "ASPNETCORE_ENVIRONMENT": "Development"
}
},
"WebApplication1": {
    "commandName": "Project",
    "dotnetRunMessages": "true",
    "launchBrowser": true,
    "launchUrl": "swagger",
    "applicationUrl":
        "https://localhost:5001;http://localhost:5000",
    "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
    }
}
}
```

The example above shows that we have settings for **IIS** and self-hosted applications (Kestrel).

If you try to check your Visual Studio, you'll see both of these profiles.

```
1  "profiles": {  
2     "IIS Express": {  
3         "commandName": "IISExpress",  
4         "launchBrowser": true,  
5         "launchUrl": "swagger",  
6         "environmentVariables": {  
7             "ASPNETCORE_ENVIRONMENT": "Development"  
8         }  
9     },  
10    "WebApplication1": {  
11        "commandName": "Project",  
12        "dotnetRunMessages": true,  
13        "launchBrowser": true,  
14        "launchUrl": "swagger",  
15        "applicationUrl": "https://localhost:5001;http://localhost:5000",  
16        "environmentVariables": {  
17            "ASPNETCORE_ENVIRONMENT": "Development"  
18        }  
19    }  
20 }
```

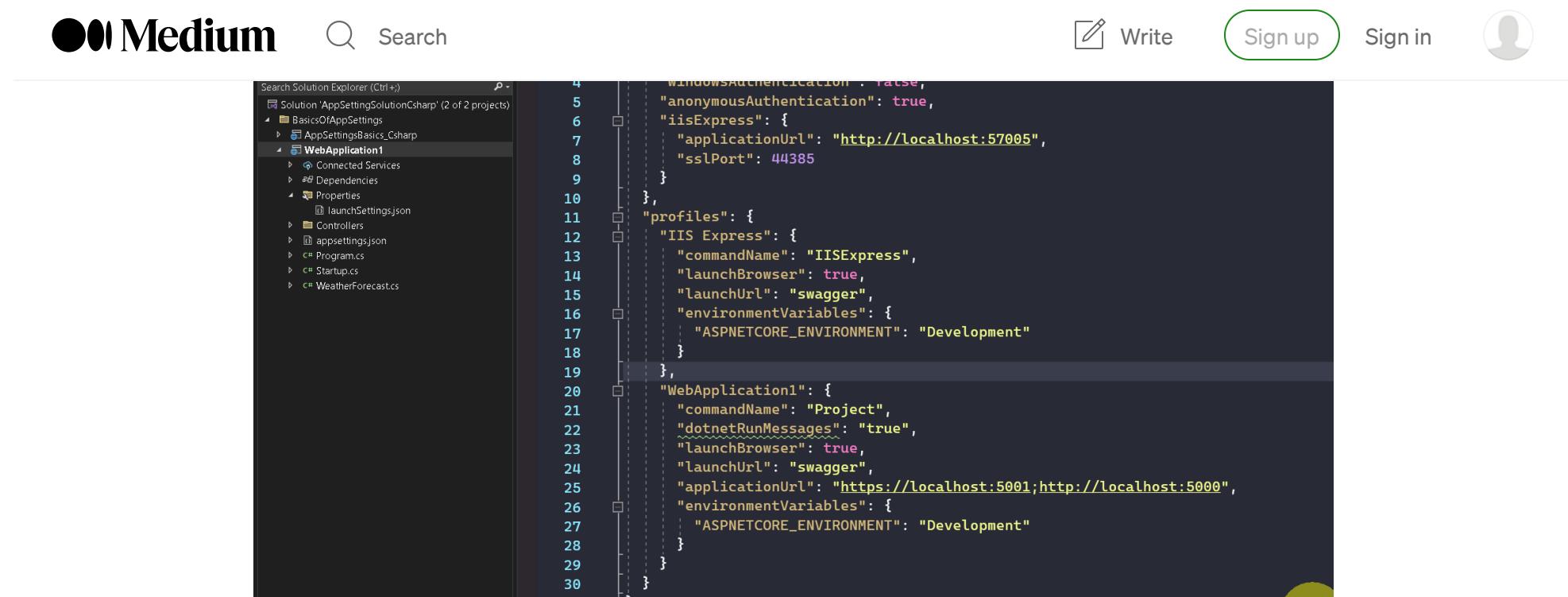
launchSettings.json profiles

Going to the profiles sub-sections like *launchBrowser*, *launchBrowser*, *applicationUrl*, etc.

You can easily change the values by just typing directly into the file with the rig us machine.

If you think you're not comfortable with that. You can use the "Launch Profile" to direct put or set the values of the properties offered by each profile. By going to "Debug Properties".

See the short video clip below.



The screenshot shows a Medium article page with a dark theme. On the left, there's a sidebar with a search bar and user navigation (Write, Sign up, Sign in). The main content area displays a code editor with the following JSON configuration:

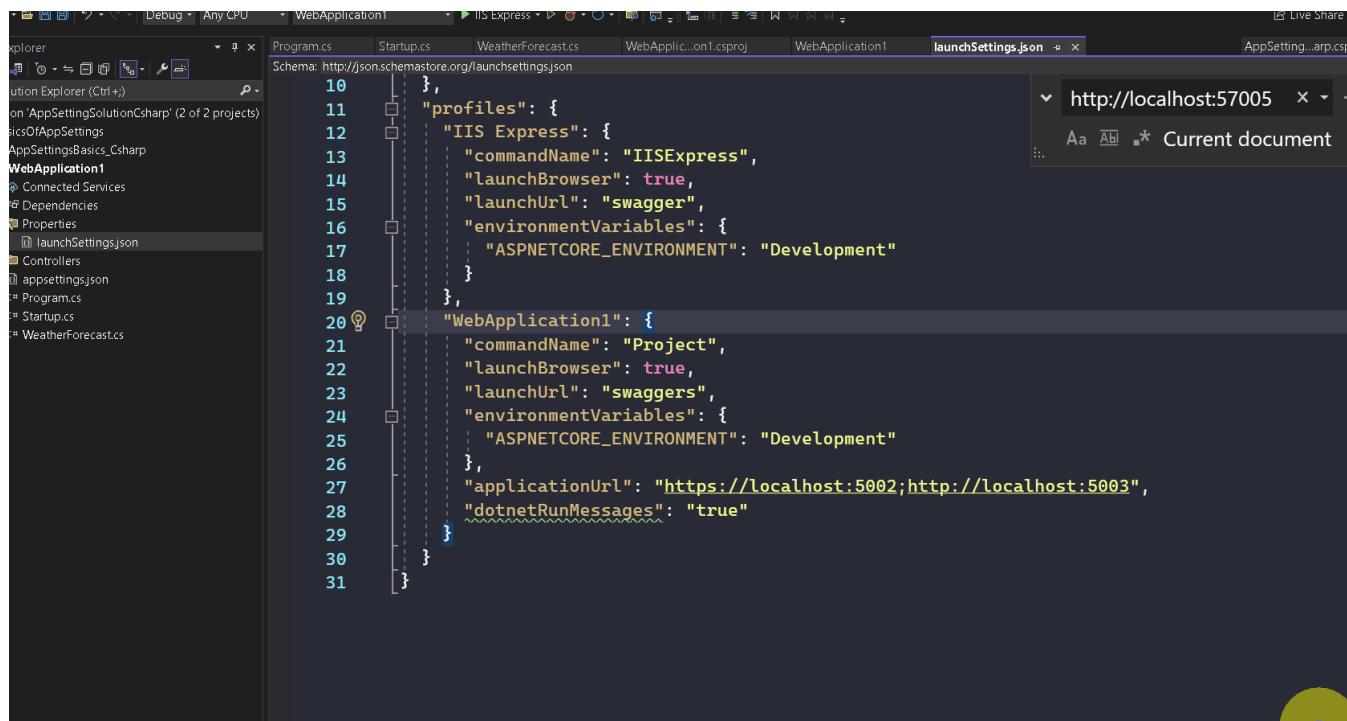
```
4      "windowsAuthentication": false,
5      "anonymousAuthentication": true,
6      "iisExpress": {
7          "applicationUrl": "http://localhost:57005",
8          "sslPort": 44385
9      },
10     },
11     "profiles": {
12         "IIS Express": {
13             "commandName": "IISExpress",
14             "launchBrowser": true,
15             "launchUrl": "swagger",
16             "environmentVariables": {
17                 "ASPNETCORE_ENVIRONMENT": "Development"
18             }
19         },
20         "WebApplication1": {
21             "commandName": "Project",
22             "dotnetRunMessages": "true",
23             "launchBrowser": true,
24             "launchUrl": "swagger",
25             "applicationUrl": "https://localhost:5001;http://localhost:5000",
26             "environmentVariables": {
27                 "ASPNETCORE_ENVIRONMENT": "Development"
28             }
29         }
30     }
31 }
```

OK now that we know that we can change those values. How about running

Vincent
To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy,
including cookie policy.
ea

How about running the project using the dotnet CLI and passing the launch profile? You can type the following command below.

```
dotnet run --launch-profile "YourProfileName"
```



```
10 },
11     "profiles": {
12         "IIS Express": {
13             "commandName": "IISExpress",
14             "launchBrowser": true,
15             "launchUrl": "swagger",
16             "environmentVariables": {
17                 "ASPNETCORE_ENVIRONMENT": "Development"
18             }
19         },
20         "WebApplication1": {
21             "commandName": "Project",
22             "launchBrowser": true,
23             "launchUrl": "swaggers",
24             "environmentVariables": {
25                 "ASPNETCORE_ENVIRONMENT": "Development"
26             },
27             "applicationUrl": "https://localhost:5002;http://localhost:5003",
28             "dotnetRunMessages": "true"
29         }
30     }
31 }
```

I made a mistake by typing swaggers on the profile but I was making a point

that To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

However, with the last short video clip that you have seen, we can run the dotnet core application using the CLI and pass the proper launch profile. Then lastly we have seen how the web-kestrel behaves when launched on the command line.

Till next time. See yeah. Cheers and bye.

Dotnet

Dotnet Core

Dotnet Core Development

Launchsettings

Aspnetcore



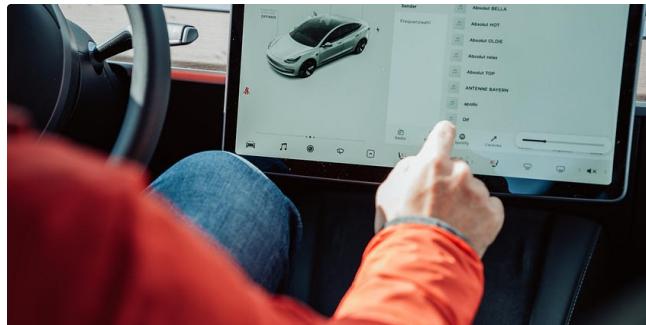
Written by Jin Vincent Necessario

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, 113 including cookie policy.

<https://www.jinoncode.dev/> - For sharing software engineering knowledge, see you there!



More from Jin Vincent Necessario



 Jin Vincent Necessario

Using IConfiguration Inside Console App .NET Core

Introduction

3 min read · Apr 2, 2023



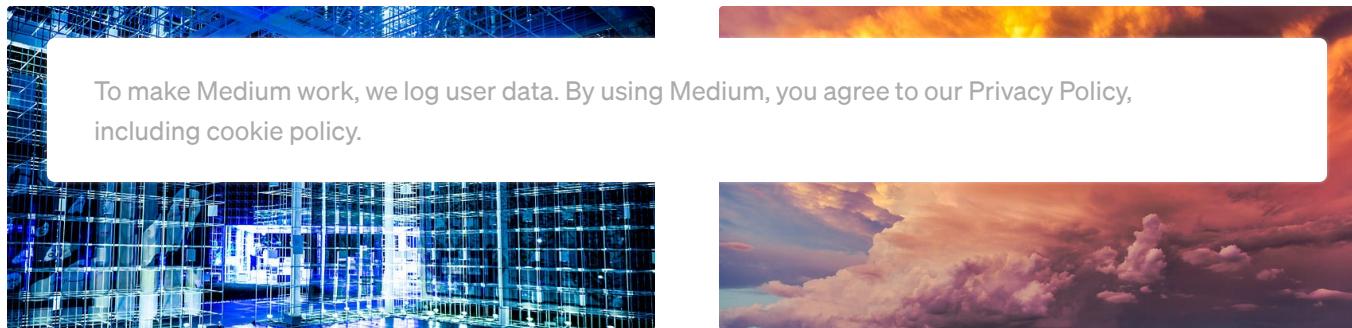
 Jin Vincent Necessario

Redis EXISTS, GET, and SET via C#

In this article, we'll explore the primary usage of Redis by adding, getting a key/value pair,...

5 min read · Apr 10, 2023





To make Medium work, we log user data. By using Medium, you agree to our [Privacy Policy](#), including cookie policy.



Jin Vincent Necessario

How to Connect to Redis via C#?

Connecting To Redis Server

2 min read · Apr 3, 2023



1



+



26

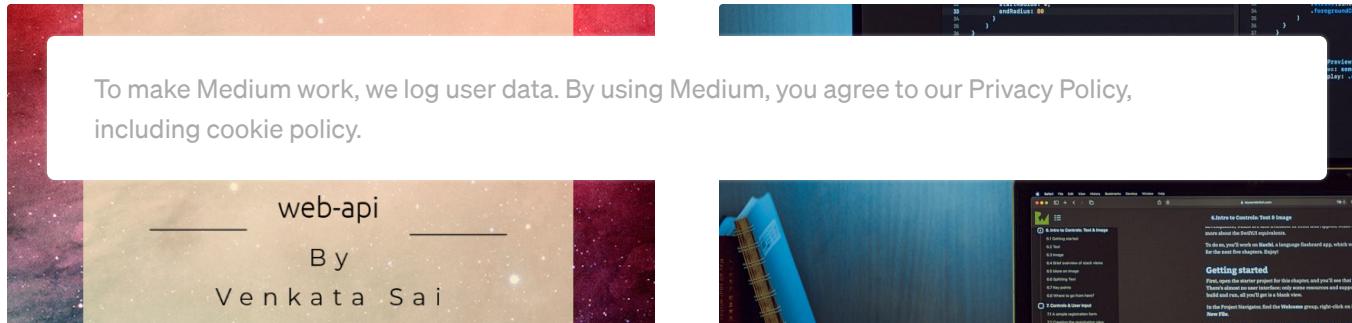


2



[See all from Jin Vincent Necessario](#)

Recommended from Medium



 Saisiva

How to configure Postgres database for a .net

In this article, we will walk through the steps to configure a PostgreSQL database for...

3 min read · Aug 7, 2023



Lists



Staff Picks

552 stories · 617 saves



Stories to Help You Level-Up at Work

19 stories · 407 saves



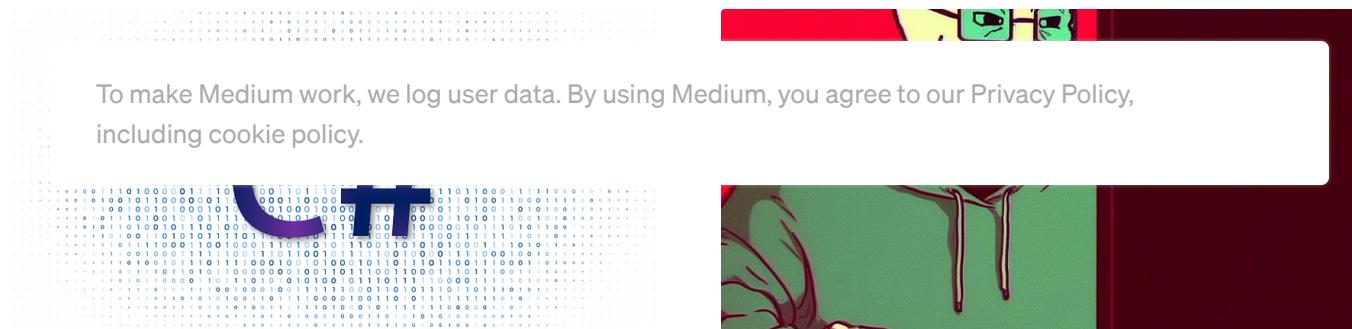
Self-Improvement 101

20 stories · 1170 saves



Productivity 101

20 stories · 1072 saves




Juldhais Hengkyawan

Upload and Save Excel File Data into the Database in ASP.NET Cor...

This article will explore the step-by-step process of uploading an Excel file and savin...

5 min read · Aug 8, 2023

 44
  2



Crafting-Code in Stackademic

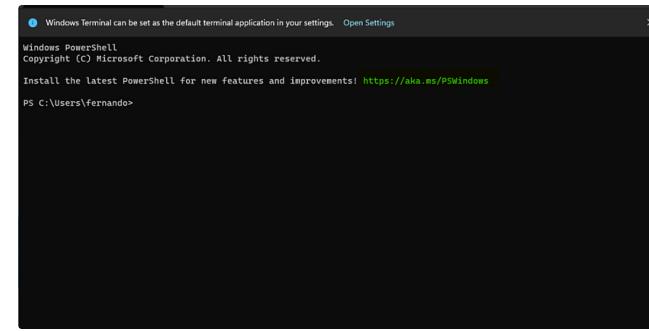



Seyyid Yiğit in Huawei Developers

C# 12 Features: A New Era of Simplicity and Elegance

C# 12 introduces some new features to make the language simpler and more elegant whil...

7 min read · Oct 17, 2023

 240
  3



Fernando Silva

What's the process for handling

Ia To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Upvotes in last 24 hours: 116

7 min read · Aug 20, 2023



116



2

[See more recommendations](#)

Windows 11 Windows Terminal

Windows Terminal application at the moment.

6 min read · Sep 8, 2023



6

