Contents

[1. Set Environment Variable to Development: 2](#_Toc204841650)

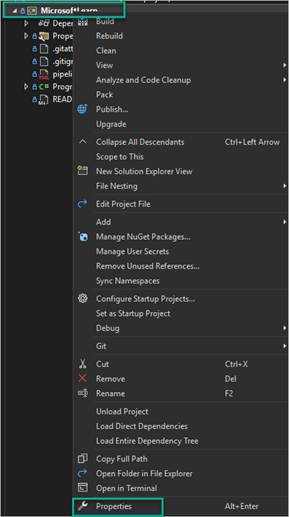
[2. Add connectionstring to appsettings.Development.json which has been moved to the Infrastructure folder, to support Clean Architecture Principles: 5](#_Toc204841651)

[3. Point Program.cs to appsettings.Development.json in the Infrastructure/Configurations folder: 6](#_Toc204841652)

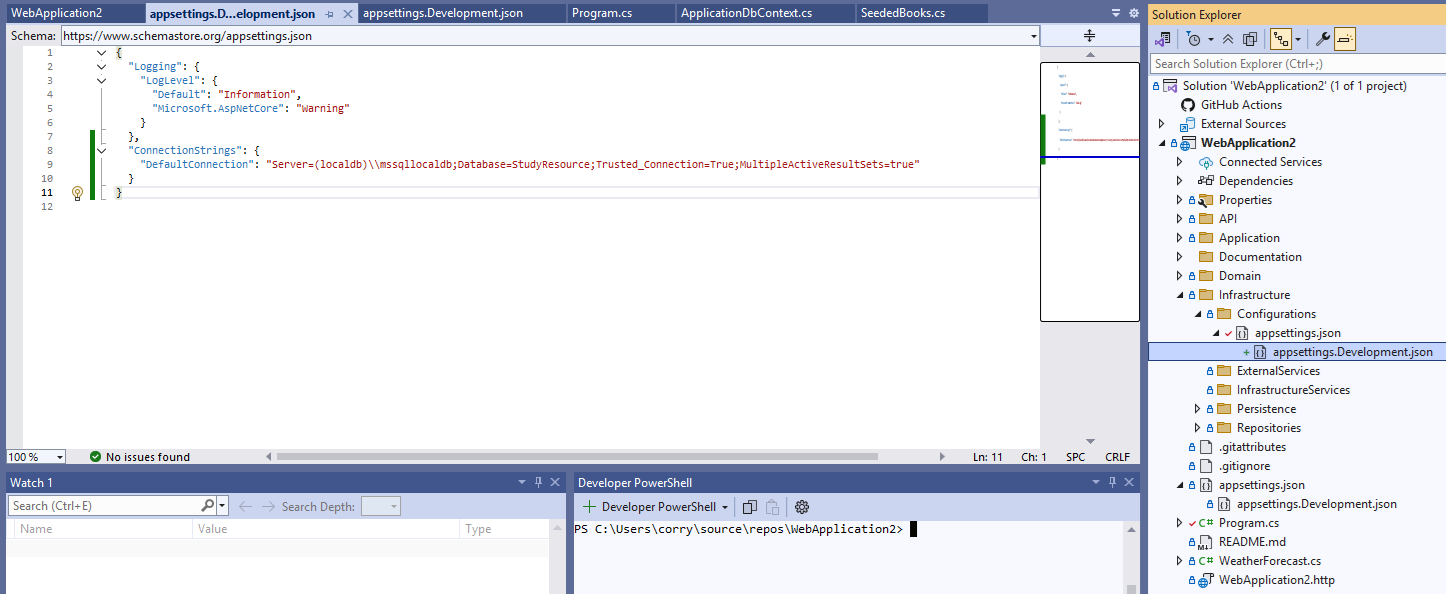
[APPENDIX: 8](#_Toc204841653)

[YouTube Video: How to keep appsettings.json out of GitHub 8](#_Toc204841654)

# Set Environment Variable to Development:

1. https://www.cloudproinc.com.au/index.php/2025/04/18/adding-environment-variables-to-visual-studio-2022/
2. **Adding Environment Variables to Visual Studio 2022**
3. In this Visual Studio and .NET blog post, we will show how to use and add environment variables in Visual Studio 2022 without adding them to the local computer or shell.
4. **Listen to this blog post**
5. By managing environment variables within Visual Studio 2022, you can streamline your development process, avoid potential conflicts, and maintain a cleaner system configuration.
6. [Visual Studio 2022,](https://www.cloudproinc.com.au/index.php/2024/07/04/streamline-your-visual-studio-updates-with-microsoft-intune/) the latest iteration of Microsoft’s premier integrated development environment (IDE), brings a plethora of features designed to streamline the development process and enhance productivity.
7. From its powerful code editor to sophisticated debugging tools and support for a wide range of programming languages, Visual Studio 2022 stands out as an essential tool for developers worldwide.
8. One of the critical functionalities that can significantly impact your development workflow is the ability to manage environment variables directly within the IDE.
9. **Why Use Environment Variables in Visual Studio?**
10. Environment variables are key-value pairs that can influence the behavior of applications. By using environment variables, developers can enhance the security, portability, and manageability of their applications.
11. **Benefits of Using Environment Variables**
12. Security: Environment variables keep sensitive information out of the source code, reducing the risk of accidental exposure.
13. Portability: Applications can be easily moved across different environments (development, testing, production) without changing the code.
14. Manageability: Configuration settings can be updated without modifying the application, allowing for easier maintenance and deployment.
15. **Adding Environment Variables to Visual Studio 2022**
16. To add environment variables in Visual Studio 2022, follow these steps:
17. **Step 1: Open the Project Properties**
18. Begin by opening your project in Visual Studio 2022. Right-click on the project name in the Solution Explorer and select “Properties” from the context menu.
19. 
20. **Step 2: Navigate to Debug Tab**
21. In the project properties window, navigate to the “Debug” tab. This tab contains settings related to debugging and launching the application.
22. **Step 3: Add Environment Variables**
23. Within the “Debug” tab, you will find a section labeled “Environment Variables.” Click the “Add” button to create a new environment variable. Enter the name and value of the variable in the respective fields.
24. ![Add Environment Variables](screenshots/add-environment-variables.png)
25. **Step 4: Apply and Save Changes**
26. After adding the necessary environment variables, click the “Apply” button to save your changes. You can now close the project properties window.
27. **Testing Environment Variables**
28. To verify that the environment variables are correctly set, you can add code to your application that reads and outputs the values of these variables. For example, in a .NET application, you can use the `Environment.GetEnvironmentVariable` method:
29. string myVariable = Environment.GetEnvironmentVariable("MY\_VARIABLE");
30. Console.WriteLine($"The value of MY\_VARIABLE is: {myVariable}");
31. Run the application in debug mode to see the output and confirm that the environment variables are working as expected.
32. **Conclusion**
33. Managing environment variables in Visual Studio 2022 is a straightforward process that can greatly enhance the security, portability, and manageability of your applications. By following the steps outlined in this guide, you can easily configure environment variables and leverage their benefits in your development workflow. Visual Studio 2022, with its extensive feature set and user-friendly interface, continues to be an indispensable tool for developers, helping them to build high-quality software efficiently.

# Add connectionstring to appsettings.Development.json which has been moved to the Infrastructure folder, to support Clean Architecture Principles:



appsettings.json

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

appsettings.Development.json

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

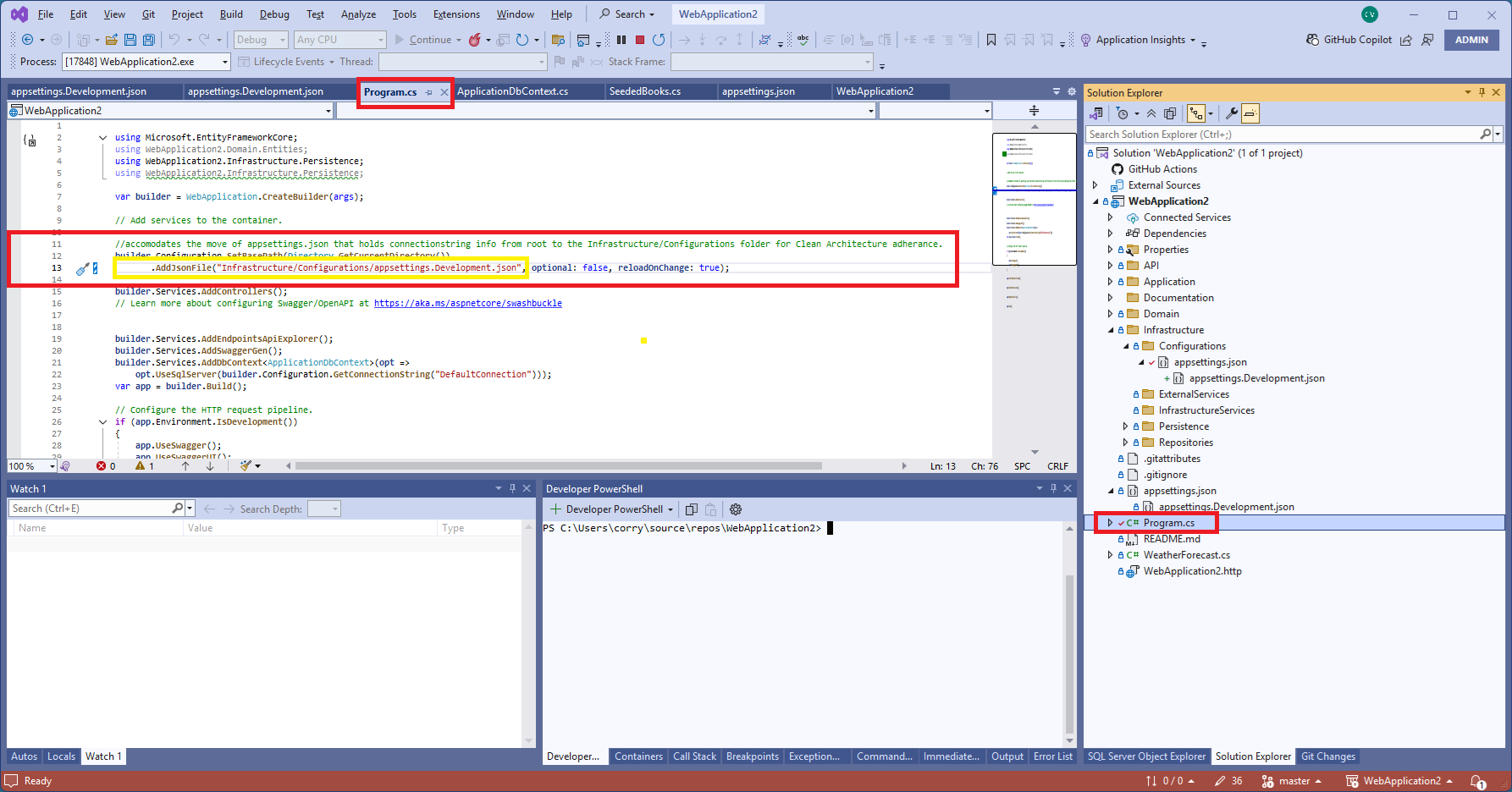
"ConnectionStrings": {

"DefaultConnection": "<connectionstring>"

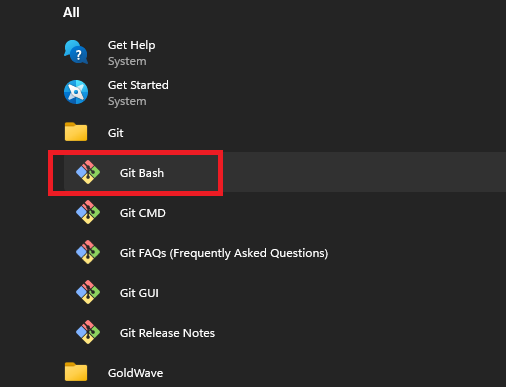
}

}

# Point Program.cs to appsettings.Development.json in the Infrastructure/Configurations folder:



# Fixing our GitHub repo if we have already committed appsettings.json and appsettings.Development.json to GitHub

1. Open Git Bash from Windows All programs
2. 
3. Navigate to the WebApplication2 directory (note the front slashes are used, not back slashes):
4. A computer screen shot of a computer program

   AI-generated content may be incorrect.
5. Enter the following Git command:

git rm -r --cached .

1. After running the command I received the following:

$ git rm -r –cached .

fatal: detected dubious ownership in repository at 'C:/Users/corry/source/repos/WebApplication2'

'C:/Users/corry/source/repos/WebApplication2' is owned by:

BUILTIN/Administrators (S-1-5-32-544)

but the current user is:

DESKTOP-3H07OJF/corry (S-1-5-21-3945321670-1450793062-2044786332-1001)

To add an exception for this directory, call:

git config --global --add safe.directory C:/Users/corry/source/repos/WebApplication2

corry@DESKTOP-3H07OJF MINGW64 /c/users/corry/source/repos/webapplication2

$ ^C

corry@DESKTOP-3H07OJF MINGW64 /c/users/corry/source/repos/webapplication2

$ ^[[200~git config --global --add safe.directory C:/Users/corry/source/repos/WebApplication2

bash: $'\E[200~git': command not found

corry@DESKTOP-3H07OJF MINGW64 /c/users/corry/source/repos/webapplication2

$ git config --global --add safe.directory C:/Users/corry/source/repos/WebApplication2

1. Then I ran the command again and Git Bash show me the caches that the command has removed:

$ git rm -r --cached .

rm 'Application/Commands/CreateResourceBookCommand.cs'

rm 'Application/DTOs/ResourceBookDTO.cs'

rm 'Application/DTOs/ResourceDTO.cs'

1. sfgsfg
2. sfgsf

# APPENDIX:

## YouTube Video: How to keep appsettings.json out of GitHub

<https://www.youtube.com/watch?v=oe7nFJXWrLs&ab_channel=ChristopherBell>