Changing Environment in .NET 5.0

To change the environment in a .NET 5.0 application, you typically manage the environment settings through the ASPNETCORE\_ENVIRONMENT variable.   
This environment variable allows the app to run in different modes like Development, Staging, and Production. Here’s how you can change it:

# 1. Set Environment in launchSettings.json

If you are running the application from Visual Studio or using dotnet run, the environment can be specified in the Properties/launchSettings.json file.  
  
Example:  
{  
 "profiles": {  
 "IIS Express": {  
 "commandName": "IISExpress",  
 "environmentVariables": {  
 "ASPNETCORE\_ENVIRONMENT": "Development"  
 }  
 },  
 "MyApp": {  
 "commandName": "Project",  
 "dotnetRunMessages": true,  
 "environmentVariables": {  
 "ASPNETCORE\_ENVIRONMENT": "Development"  
 }  
 }  
 }  
}  
In this example, the ASPNETCORE\_ENVIRONMENT is set to Development for both IIS Express and the application itself.   
You can change it to Staging or Production based on your needs.

# 2. Set Environment Variable in the Operating System

Windows (Command Prompt / PowerShell)  
  
You can set the ASPNETCORE\_ENVIRONMENT environment variable before running the application:  
- Command Prompt:  
 set ASPNETCORE\_ENVIRONMENT=Production  
 dotnet run  
  
- PowerShell:  
 $env:ASPNETCORE\_ENVIRONMENT="Production"  
 dotnet run  
  
Linux/Mac (Bash):  
 export ASPNETCORE\_ENVIRONMENT=Production  
 dotnet run

# 3. Set Environment in Azure

If your app is deployed to Azure, you can set the environment variable directly in the Azure portal under App Service > Configuration > Application settings.   
Add or update the ASPNETCORE\_ENVIRONMENT variable there.

# 4. Set Environment in Docker

If you are running your application in a Docker container, you can set the environment in the Dockerfile or docker-compose.yml file using:  
  
Dockerfile:  
 ENV ASPNETCORE\_ENVIRONMENT=Production  
  
Or in docker-compose.yml:  
 environment:  
 - ASPNETCORE\_ENVIRONMENT=Production

# 5. Set Environment in IIS Manager:

1. **Open IIS Manager**
   * Open the **IIS Manager** by typing **inetmgr** in the **Run** window or by searching for it in the Start menu.
2. **Select Your Site**
   * In IIS Manager, expand the Connections pane on the left.
   * Find your server under Connections > Sites and select the specific site for which you want to set the environment.
3. **Open the Configuration Editor**
   * In the middle pane, locate and open the **Configuration Editor** feature.
     + If you don't see this, make sure you are selecting your site in the **Connections** pane.
4. **Navigate to Environment Variables**
   * In the **Configuration Editor**, click on the dropdown menu (under **Section**) and navigate to:
     + system.webServer/aspNetCore/environmentVariables
5. **Add ASPNETCORE\_ENVIRONMENT Variable**
   * In the **environmentVariables** section, click the **(Collection)** to open the environment variables editor.
   * Click **Add** and set the following:
     + **name**: ASPNETCORE\_ENVIRONMENT
     + **value**: Set this to either Development, Testing, Production, or any custom environment you use.
   * Click **OK** to save the changes.
6. **Apply Configuration**
   * After making the changes, click **Apply** on the right side of the IIS Manager to apply the configuration.
7. **Restart the IIS Site**
   * For the new environment settings to take effect, you may need to **restart** your IIS site. You can do this by right-clicking on your site in the **Connections** pane and selecting **Restart**.