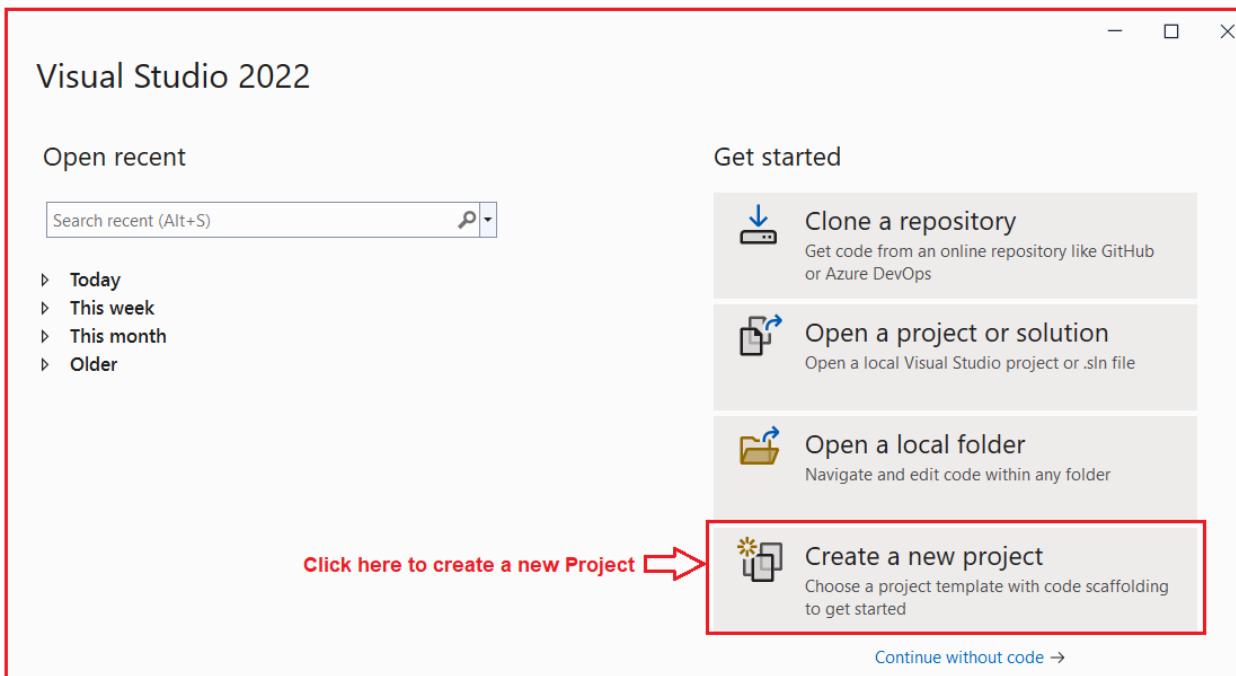
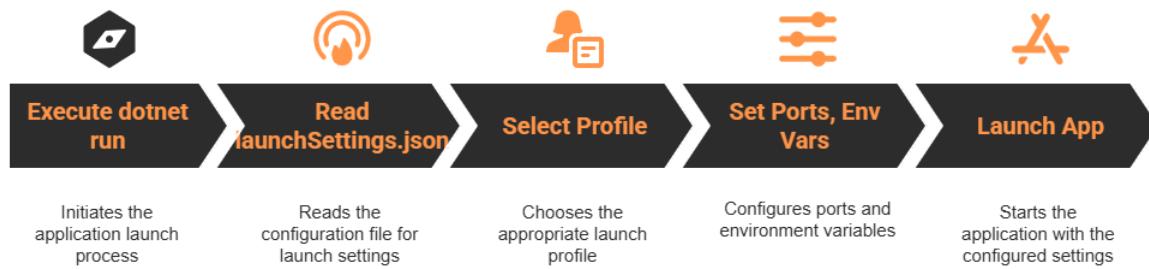
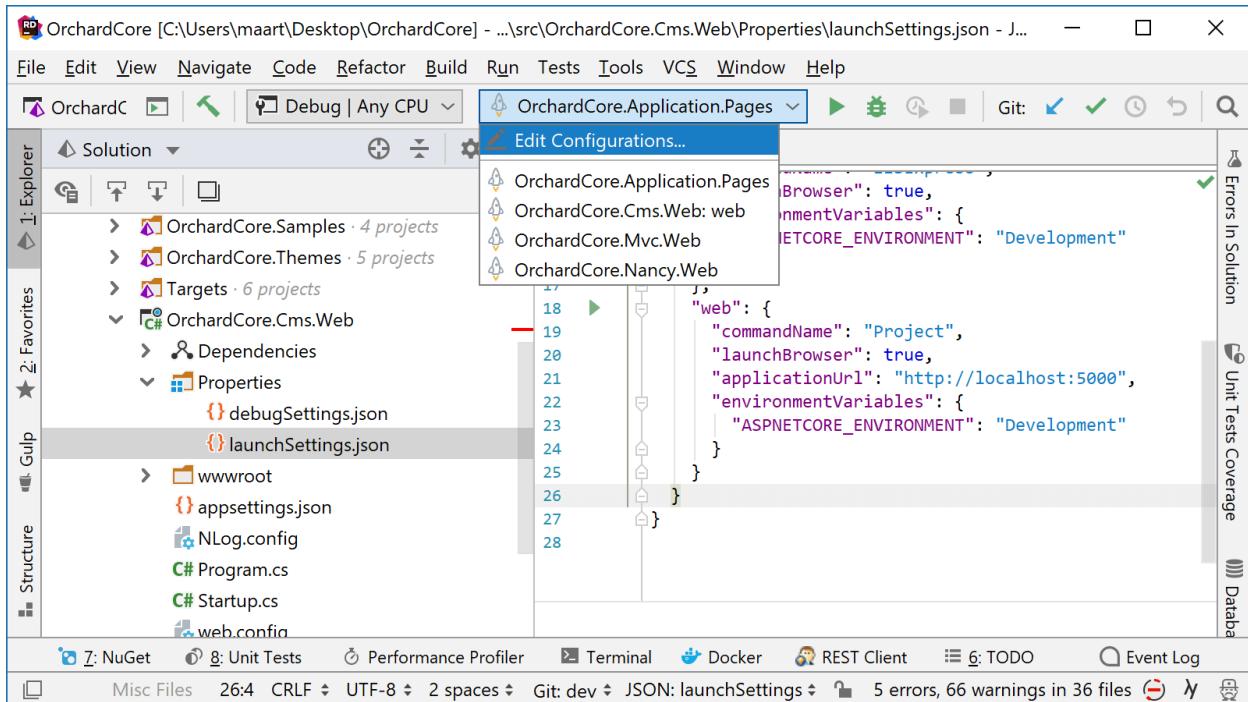


🚀 launchSettings.json – Overview



Application Launch Workflow





The screenshot shows the Visual Studio IDE interface with the title bar "OrchardCore [C:\Users\maart\Desktop\OrchardCore] - ...\\src\\OrchardCore.Cms.Web\\Properties\\launchSettings.json - J...". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tests, Tools, VCS, Window, and Help. The toolbar has icons for Debug (Any CPU), Run, Stop, and others. The Solution Explorer on the left shows the project structure with "OrchardCore.Samples", "OrchardCore.Themes", "Targets", and "OrchardCore.Cms.Web" selected. Under "OrchardCore.Cms.Web", "Properties" is expanded, showing "debugSettings.json" and "launchSettings.json". The "Properties" tab is active. The main code editor window displays the "launchSettings.json" file:

```
 18  "Browser": true,
 19  "environmentVariables": {
 20    "ASPNETCORE_ENVIRONMENT": "Development"
 21  },
 22  "web": {
 23    "commandName": "Project",
 24    "launchBrowser": true,
 25    "applicationUrl": "http://localhost:5000",
 26    "environmentVariables": {
 27      "ASPNETCORE_ENVIRONMENT": "Development"
 28    }
}
```

4

launchSettings.json is a configuration file used only **during development**.

It controls **how your ASP.NET Core application starts** when you run it locally using:

- Visual Studio
- VS Code
- dotnet run

It **does not** get published to production.

📌 Where is launchSettings.json located?

YourProject/

 └ Properties/

 └ launchSettings.json

⌚ Why do we need launchSettings.json?

It helps developers by defining:

- ✓ Application URL(s)
 - ✓ Environment (Development / Staging / Production)
 - ✓ Whether to use IIS Express or Kestrel
 - ✓ HTTP / HTTPS ports
 - ✓ Command-line arguments
 - ✓ Browser auto-launch on debugging
-

 **Example launchSettings.json**

```
{  
  "profiles": {  
    "IIS Express": {  
      "commandName": "IISExpress",  
      "launchBrowser": true,  
      "environmentVariables": {  
        "ASPNETCORE_ENVIRONMENT": "Development"  
      }  
    },  
    "MyApi": {  
      "commandName": "Project",  
      "dotnetRunMessages": true,  
      "launchBrowser": true,  
      "applicationUrl": "https://localhost:7153;http://localhost:5153",  
      "environmentVariables": {  
        "ASPNETCORE_ENVIRONMENT": "Development"  
      }  
    }  
  }  
}
```

```
}
```

✳️ Key Sections Explained

1 Profiles Section

A **profile** tells ASP.NET Core *how to run the app.*

Common profiles:

- **IIS Express** (only in Visual Studio Windows)
 - **Project/Kestrel** (cross-platform)
-

2 commandName

Defines which runtime to use:

commandName Meaning

"IISExpress" Run using IIS Express

"Project" Run using Kestrel (dotnet run)

3 applicationUrl

Specifies the URLs the project should listen on:

"https://localhost:7153;http://localhost:5153"

It supports multiple URLs separated by semicolon.

4 launchBrowser

If **true**, browser opens automatically with your API/swagger.

5 environmentVariables

Most important one:

```
"ASPNETCORE_ENVIRONMENT": "Development"
```

This controls the environment your project runs in.

It is how ASP.NET Core decides to use:

- appsettings.Development.json
 - Developer Exception Page
 - Swagger (typically enabled only in dev)
-

🔥 launchSettings.json vs appsettings.json

Feature	launchSettings.json	appsettings.json
Controls how the app starts	✓ Yes	✗ No
Stores configuration values	✗ No	✓ Yes
Used in production	✗ No	✓ Yes
Contains environment variables	✓ Yes	✓ Yes
Contains URLs / profiles	✓ Yes	✗ No

💻 When does launchSettings.json load?

When you run:

dotnet run

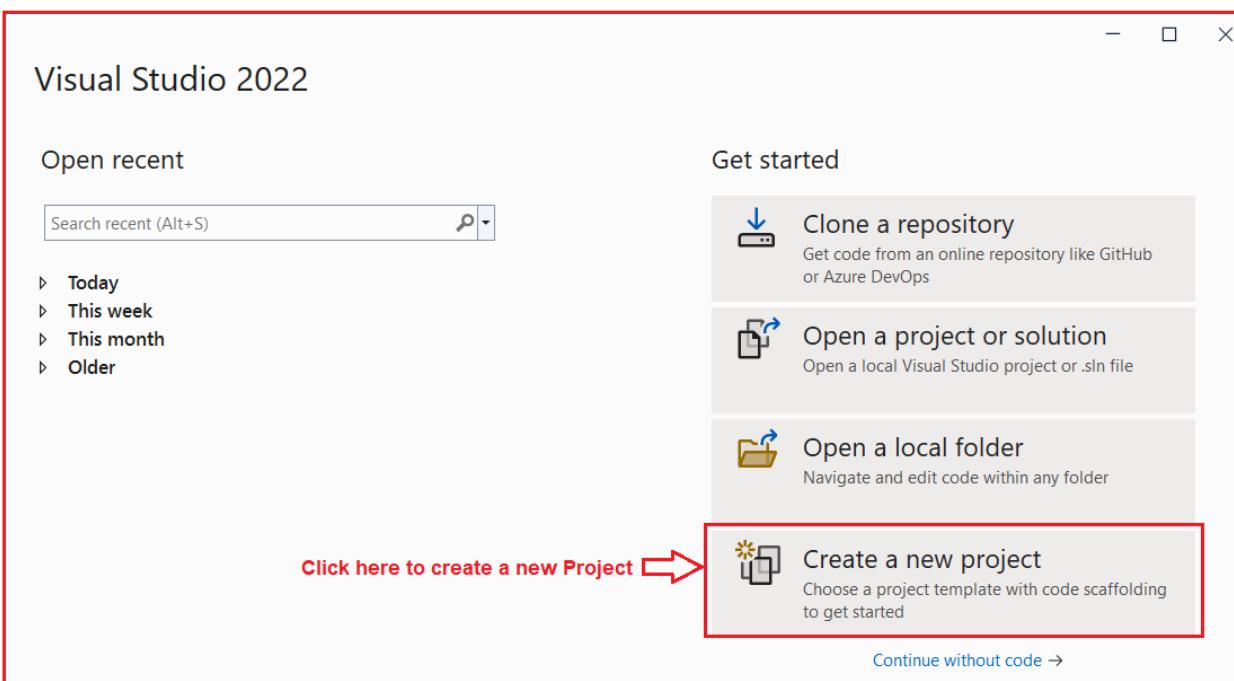
ASP.NET Core automatically loads:

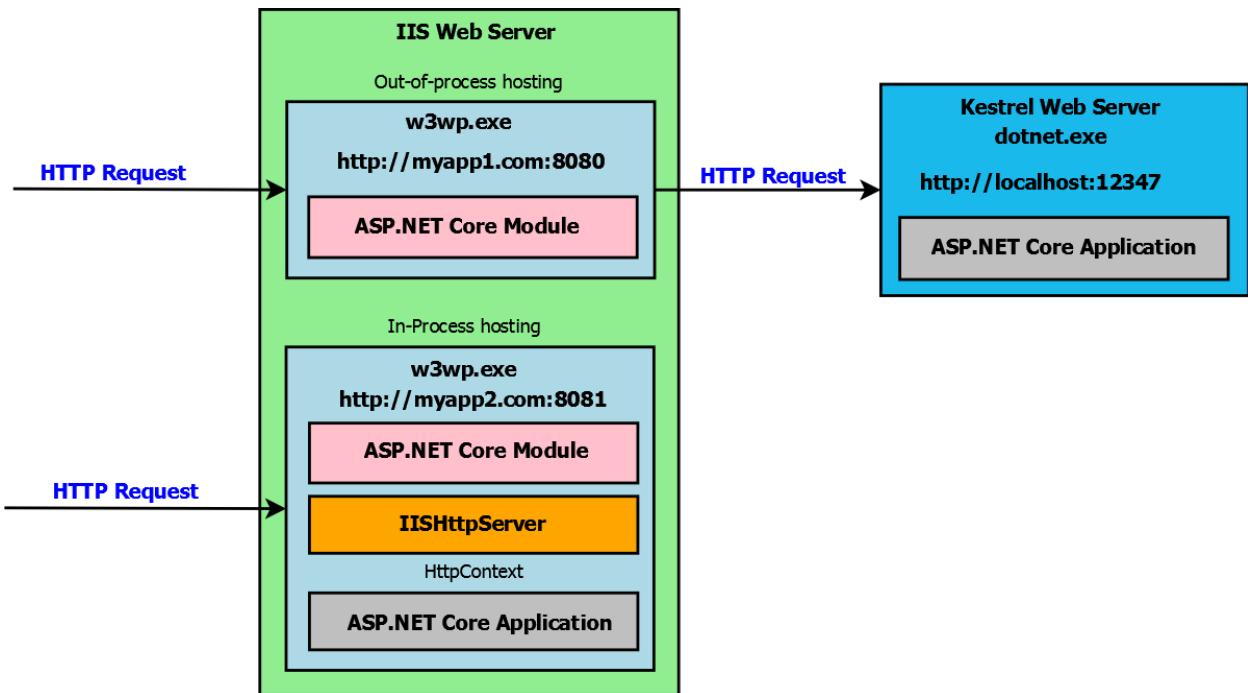
- launchSettings.json
- The profile marked as active
- Sets environment variables

But when you **publish**, this file is ignored.

🌐 Diagram – How launchSettings.json Works

Application Launch Workflow





4

🎓 Summary for Students

- `launchSettings.json` controls **how your Web API runs locally**
- It sets **URLs, environment, browser settings, profile names**
- It is **NOT used in production**
- It tells ASP.NET Core which environment to run (Development/Staging/Production)
- It makes local development easier and predictable