

Here's the **step-by-step guide**:

Hosting ASP.NET Core Web API with IIS

1. Install the .NET Core Hosting Bundle

- On the Windows Server, download and install the **ASP.NET Core Hosting Bundle**: [Download .NET Hosting Bundle](#)
 - This installs:
 - .NET Runtime
 - ASP.NET Core Module (ANCM) for IIS (used to forward requests to Kestrel).
-

2. Publish the Web API

On your development machine (or server), run:

```
dotnet publish -c Release -o ./publish
```

This generates the published files inside the **publish** folder, including:

- **MyWebApi.dll**
 - Dependencies
 - **web.config** (important for IIS)
-

3. Copy Published Files to the Server

Copy the contents of the **publish** folder to a directory on the server, for example:

```
C:\inetpub\wwwroot\MyWebApi
```

4. Configure the IIS Website

1. Open **IIS Manager** (**inetmgr**).
2. Right-click **Sites** → **Add Website...**
 - **Site name**: **MyWebApi**
 - **Physical path**: **C:\inetpub\wwwroot\MyWebApi**
 - **Binding**: Select **http** (port **80**), or **https** (port **443**) if you want SSL.

3. Click **OK**.

5. Verify `web.config`

The publish process generates a `web.config` file like this:

```
<configuration>
  <system.webServer>
    <handlers>
      <add name="aspNetCore" path="*" verb="*"
          modules="AspNetCoreModuleV2"
          resourceType="Unspecified" />
    </handlers>
    <aspNetCore processPath="dotnet"
        arguments=".\MyWebApi.dll"
        stdoutLogEnabled="false"
        stdoutLogFile=".\logs\stdout"
        hostingModel="OutOfProcess" />
  </system.webServer>
</configuration>
```

⚡ Note:

- `processPath="dotnet"` means IIS will use the installed runtime.
- If you published as **self-contained**, this would instead point directly to your app executable.

6. Configure Application Pool

1. In **IIS Manager** → **Application Pools**:

- Select the app pool created for your API.
- Set **.NET CLR Version** → **No Managed Code** (since ASP.NET Core doesn't use IIS's pipeline).
- Ensure the app pool identity has permission to access your folder.

7. Configure Environment Variables (Optional)

If you want to specify environment (`Development`, `Staging`, `Production`):

- Add `ASPNETCORE_ENVIRONMENT` to **System Environment Variables** or in `web.config`:

```
<aspNetCore processPath="dotnet" arguments=".\MyWebApi.dll">
  <environmentVariables>
    <environmentVariable name="ASPNETCORE_ENVIRONMENT" value="Production" />
  </environmentVariables>
</aspNetCore>
```

8. Restart IIS and Browse

- Restart IIS:

```
iisreset
```

- Open your API in the browser or via Postman:

```
http://localhost/api/values
```

☒ Summary

1. Install Hosting Bundle.
 2. Publish Web API.
 3. Copy to `inetpub\wwwroot`.
 4. Add IIS Website.
 5. Ensure `web.config` points to your `.dll`.
 6. Set App Pool → No Managed Code.
 7. Restart IIS.
 8. Access API via `http://yourserver/api/....`
-