

# Get .NET Core Web Api Hosted in IIS

**Hongwei Li**



Senior Developer  
Chicago, IL, USA  
2017-11-23

When I created .NET Core Web Apis, I can deploy them in Microsoft Azure portal, then they are hosted in Azure environment and they can work very well. But when I deployed them in my local IIS (Internet Information Services), the Web Apis cannot work. My article will help you solve this problem. This approach in my article can solve the same problem in Windows 7, Windows 8 and Windows 10 workstations and Servers.

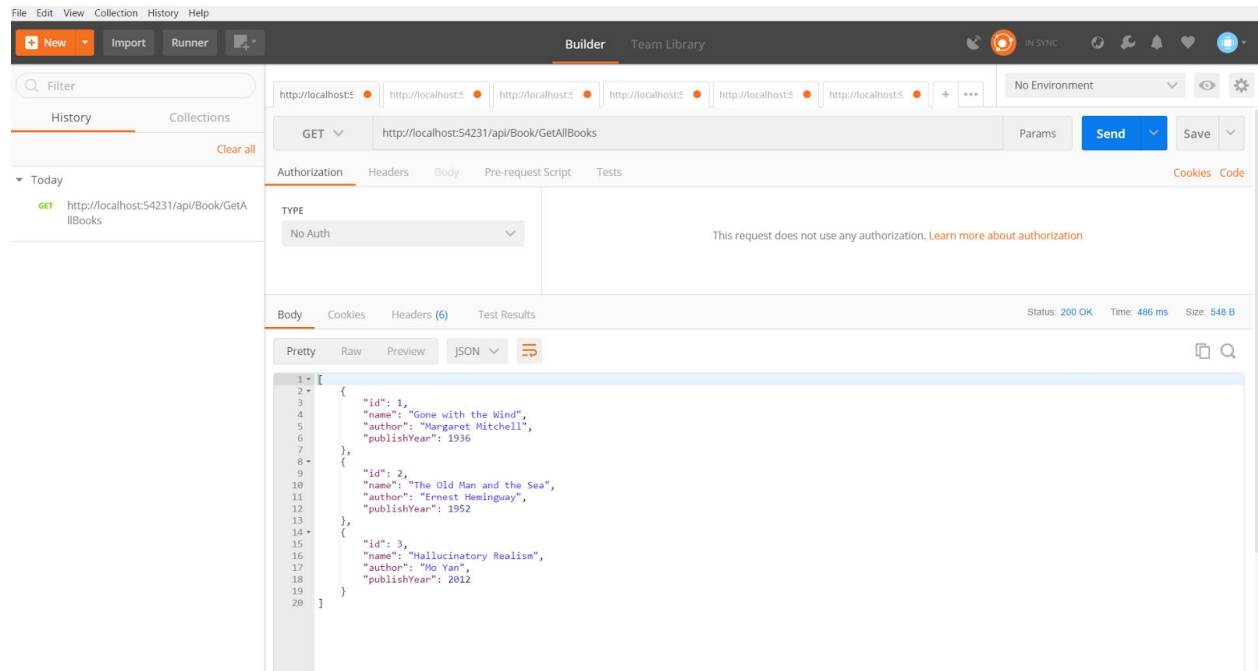
My test system information is as follows:

- Visual Studio Community 2017
- .NET Core 2.0
- Windows 10 Professional
- Internet Information Services version 10
- Postman

## **1. Develop the Web Apis**

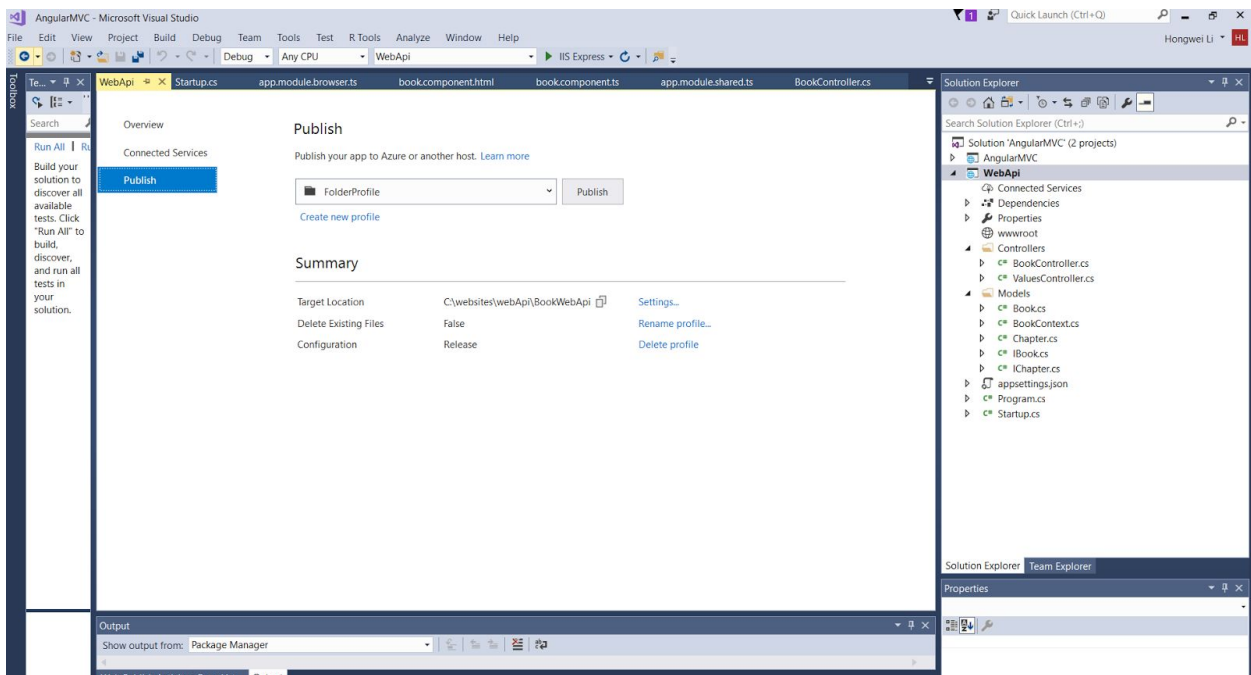
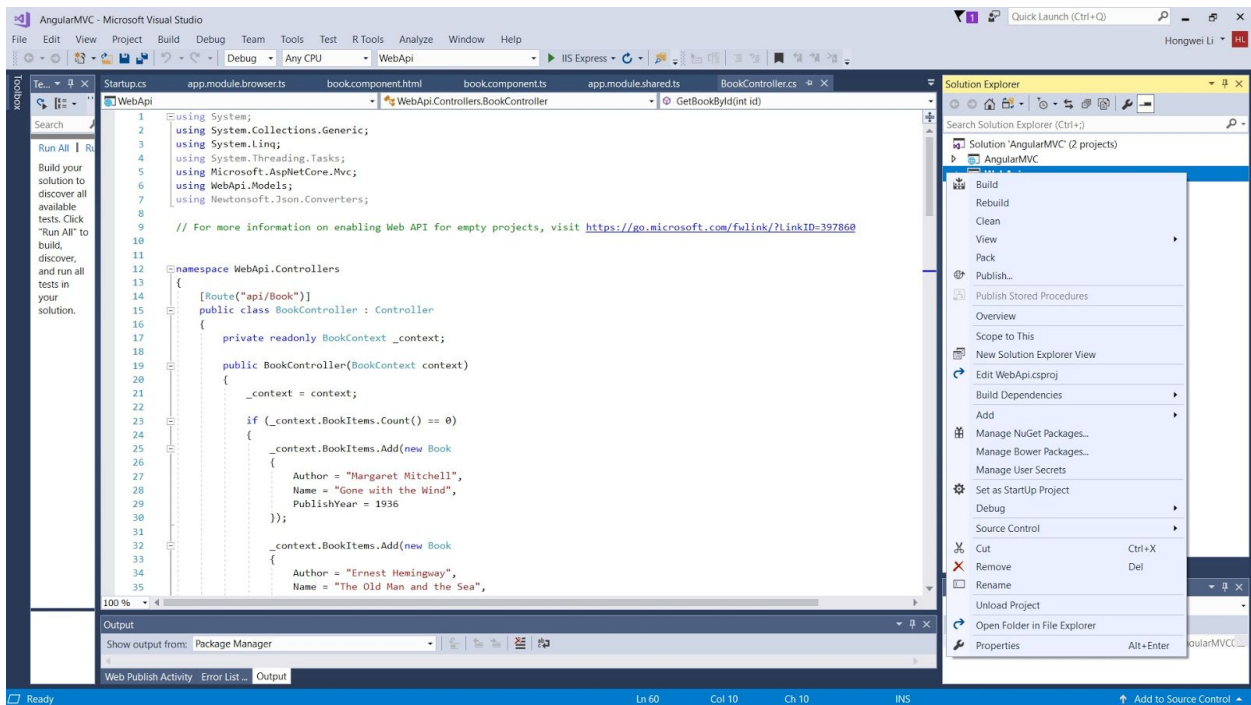
When you finish developing the Web Apis under .NET Core in Visual Studio Community 2017, you need to test to make sure your Web Apis are working in Postman.

Press F5 to Start your Web Api project in Visual Studio. Run Postman with your proper Web Api urls to test to make sure your Web Apis are working.



## 2. Publish the Web Apis to your local folder

You can highlight your Web Api project in your solution in Visual Studio and right mouse click “Publish...” to publish your Web Api project to your assigned local folder, for example, C:\websites\webApi\BookWebApi .



### 3. Install Microsoft .NET Core Windows Server Hosting bundle file

You can search the web for “.NET Core Windows Server Hosting bundle” file ( At my time, it is DotNetCore.2.0.3-WindowsHosting.exe ). Install this file in your work station.

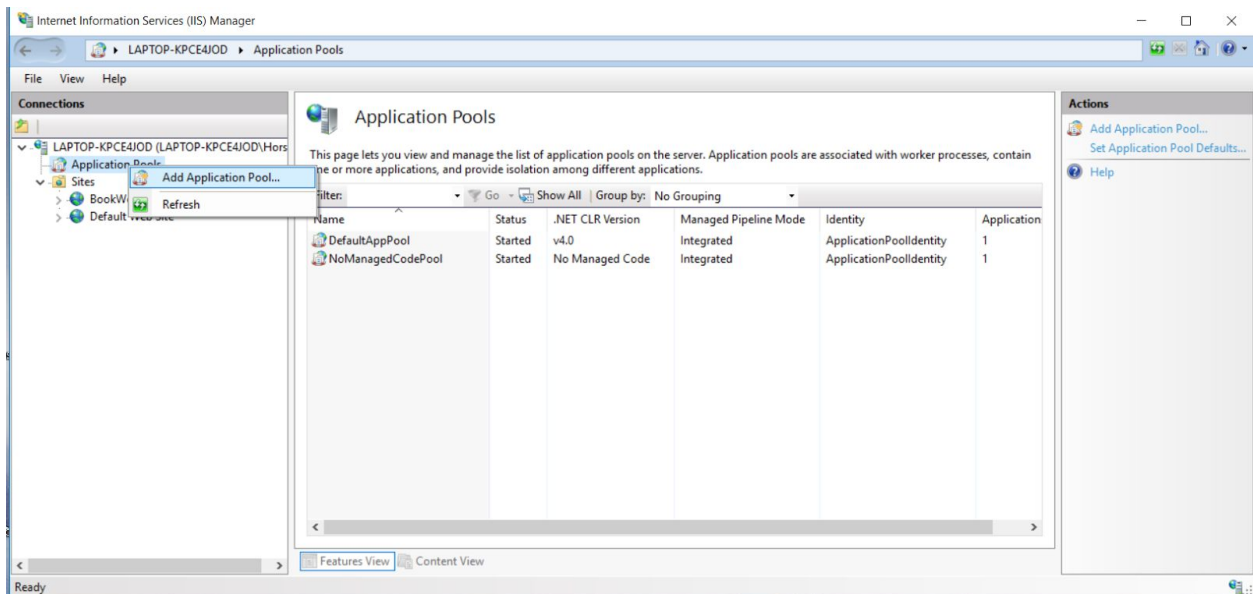
## 4. Configure Internet Information Services

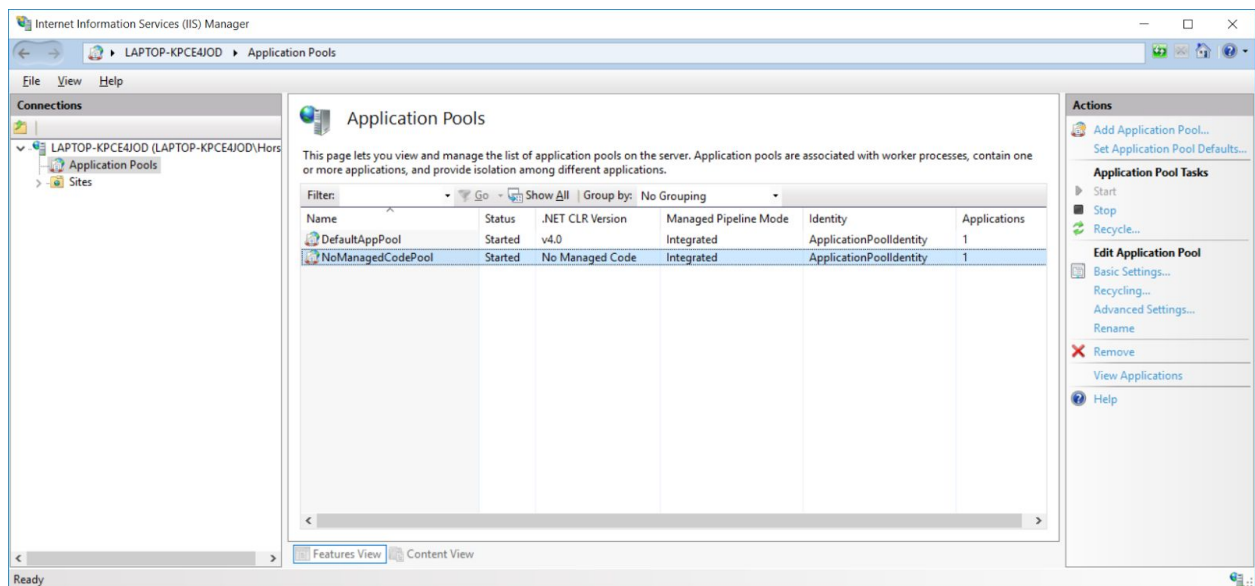
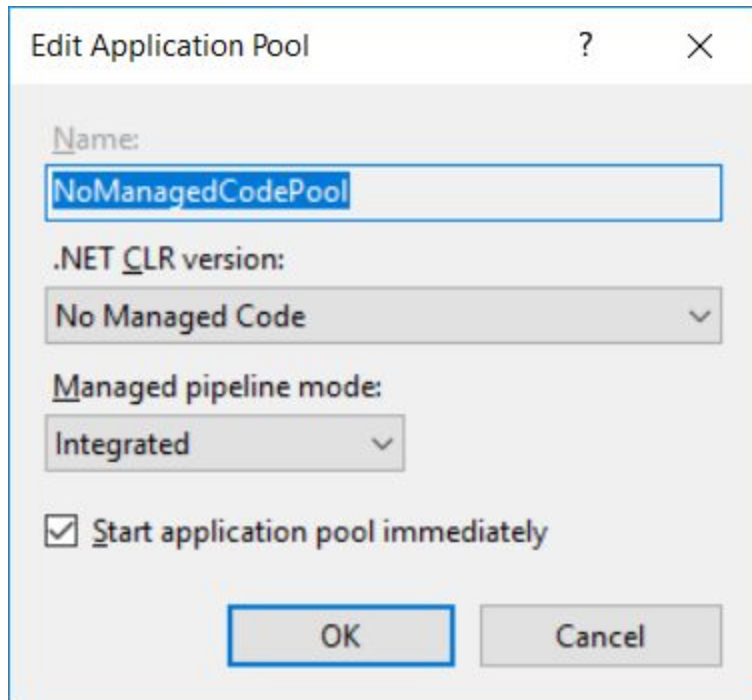
We need to understand that, per Microsoft,

“.NET Core is a general purpose development platform maintained by Microsoft and the .NET community on GitHub. It is cross-platform, supporting Windows, macOS and Linux, and can be used in device, cloud, and embedded/IoT scenarios.”

### (1) create one application pool of No Managed Code

.NET Core is not dependent on the managed code environment. Therefore, we need to create one application pool of No Managed Code at IIS. You can highlight “Application Pools” and right mouse click to “Add Application Pool...”, for example, NoManagedCodePool.





## (2) Create your web site

In IIS, you can highlight “Sites” and right mouse click to “Add Website...”, for example, BookWebApi, to point to the local folder of your published Web Api code , for example, C:\websites\webApi\BookWebApi, and use “NoManagedCodePool” with port 9001.

**Edit Site**

Site name:  Application pool:

Physical path:

Pass-through authentication

Internet Information Services (IIS) Manager

LAPTOP-KPCE4JOD > Sites

File View Help

Connections

- LAPTOP-KPCE4JOD (LAPTOP-KPCE4JOD)\Hosts
  - Application Pools
  - Sites
    - BookWebApi
    - Default Web Site

Sites

Filter:  Go  Group by: No Grouping

Name	ID	Status	Binding	Path
BookWebApi	3	Started (http)	*:9001 (http)	C:\websites\webApi\BookWebApi
Default Web Site	1	Started (http)	*:80 (http)	%SystemDrive%\inetpub\wwwroot

Features View Content View

Ready

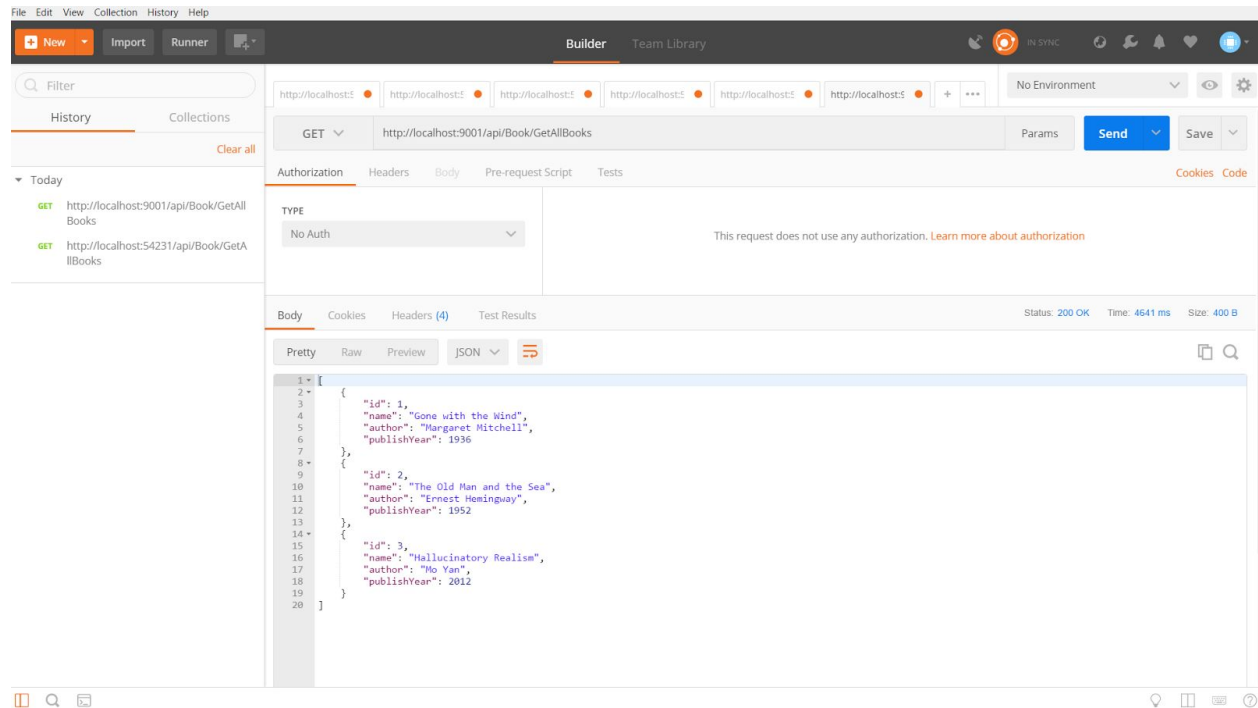
Actions

- Add Website...
- Set Website Defaults...
- Help

Start your web site for your Web Apis.

#### 4. Test your Web Apis hosted in IIS.

Now you can use Postman to test your Web Apis hosted in IIS.



Then you have successfully hosted your Web Apis developed with .NET Core in IIS locally.

Happy coding!

Hongwei Li

Senior Developer in Chicago, IL, USA