How-To: Run and debug ASP.NET Core apps in IIS Express

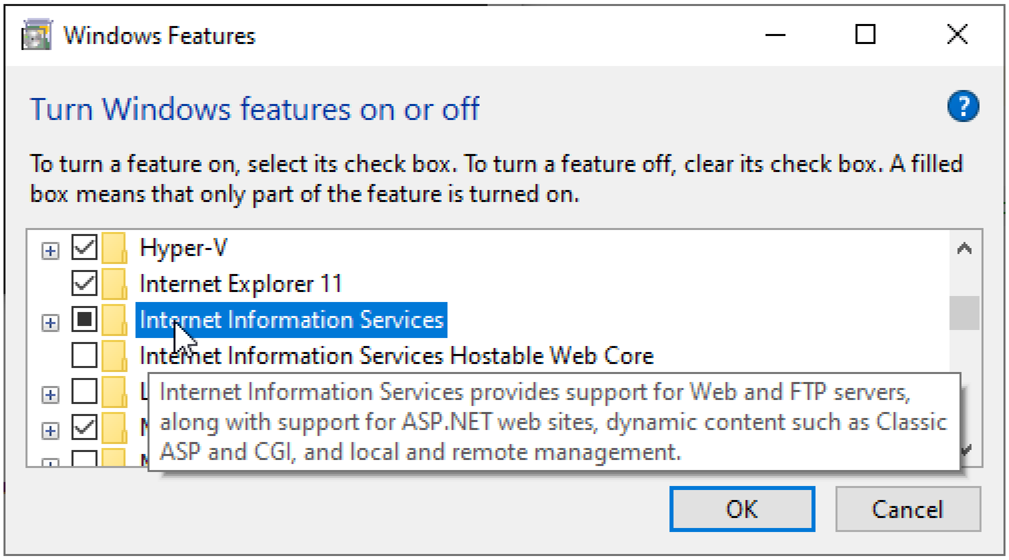
Running and Debugging ASP.NET Core Apps in IIS Express only works on Windows.

Before you start

1. Download and install IIS Express
2. To host ASP.NET Core applications, IIS Express relies on the ASP.NET Core Module. If you have Visual Studio 2019 or later on your machine, this module is already installed.

   Without Visual Studio, you need to install this module. To do so, install the required components in the following order:

   1. Internet Information Services — **Windows button | Turn Windows features on or off**.

      Windows Features

      Turn Windows features on or off

      To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

      ☑ Hyper-V
      ☑ Internet Explorer 11
      ■ Internet Information Services
      □ Internet Information Services Hostable Web Core

      Internet Information Services provides support for Web and FTP servers, along with support for ASP.NET web sites, dynamic content such as Classic ASP and CGI, and local and remote management.

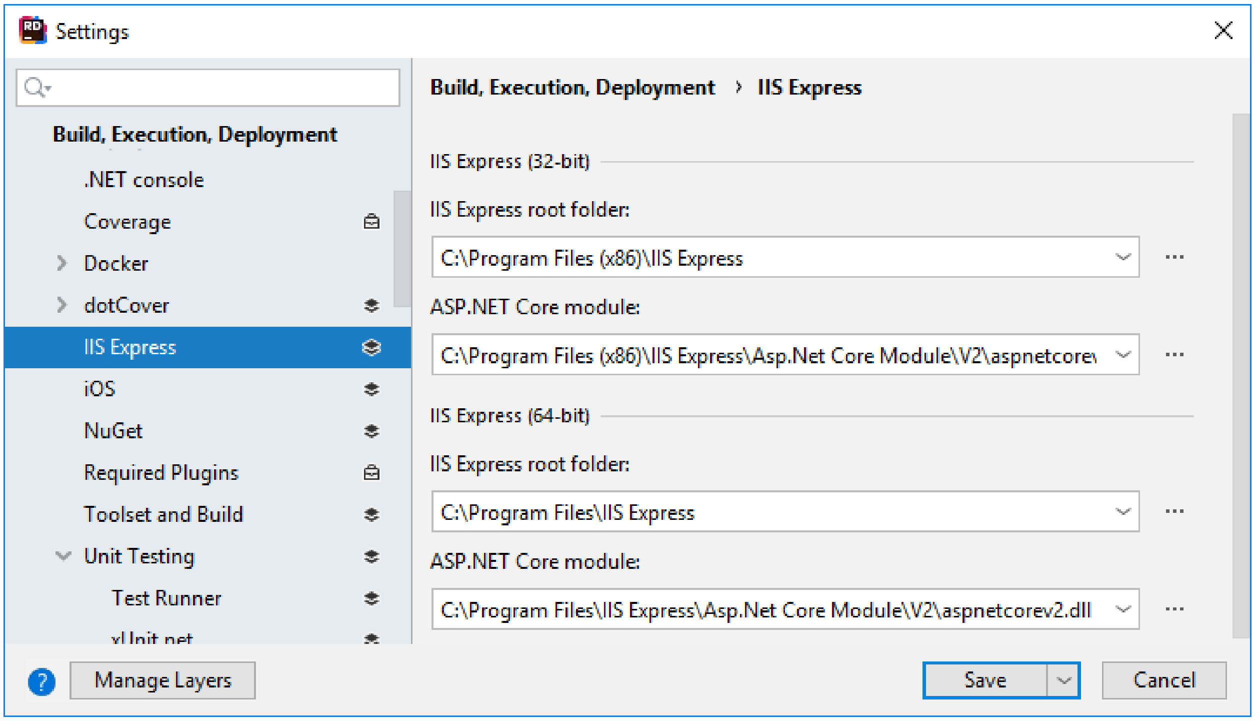
   2. .NET Core Hosting Bundle

   If you install IIS after the Hosting Bundle, you will need to run the Hosting Bundle installer to repair the installation.
3. You can check if you have the module in your IIS Express configuration file **%PROGRAMFILES(x86)%\IIS Express\config\templates\PersonalWebServer\applicationhost.config** . If there is at least one of the following lines, then you have the module installed:
4. <add name="AspNetCoreModule"...

   <add name="AspNetCoreModuleV2"...

Copied!

5. If everything is configured correctly, JetBrains Rider will automatically find all necessary components. You can check their paths on the **Build, Execution, Deployment | IIS Express** page of JetBrains Rider settings Ctrl+Alt+S:



Build, Execution, Deployment › IIS Express

IIS Express (32-bit)

IIS Express root folder: C:\Program Files (x86)\IIS Express

ASP.NET Core module: C:\Program Files (x86)\IIS Express\Asp.Net Core Module\V2\aspnetcore\

IIS Express (64-bit)

IIS Express root folder: C:\Program Files\IIS Express

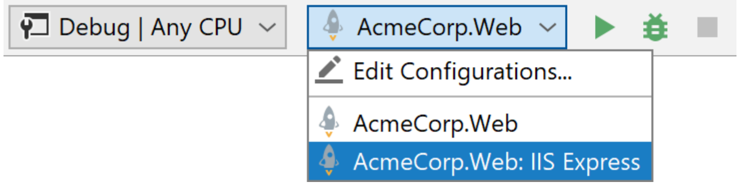
ASP.NET Core module: C:\Program Files\IIS Express\Asp.Net Core Module\V2\aspnetcorev2.dll

In case IIS Express does not start or returns an empty error, there is a chance the .NET Core Hosting Bundle did not install correctly. [This article](#) (under **Fixing the applicationhost.config template**) may help troubleshoot and fix the default IIS Express configuration template on your machine.

Run/debug configurations

After creating a new ASP.NET Core project or loading an existing one that has a [launchSettings.json](#) file with one or more IIS Express profiles, JetBrains Rider will [automatically create run/debug configurations based on those profiles](#).

You can view and edit created configurations either in the **Run/Debug Configurations** dialog (**Run | Edit Configurations**) or on the toolbar selector:



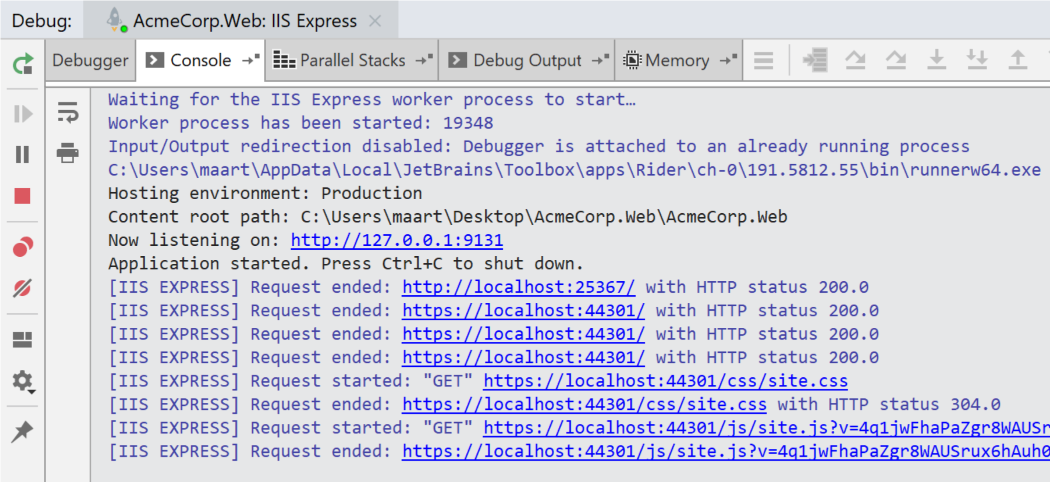
For IIS Express profiles, the run configuration includes a few additional options:

Generate applicationhost.config

To launch IIS Express with your ASP.NET Core application, JetBrains Rider uses file **.idea/config/applicationhost.config**, which is created using a template th Express. When possible, customizations you make to this file are also preserve disable this option for heavy customizations.

Show IIS Express output

This option enables output from the IIS Express process. By default, it is disable quite verbose), but when enabled we can see additional details when running



```
Waiting for the IIS Express worker process to start…
Worker process has been started: 19348
Input/Output redirection disabled: Debugger is attached to an already running process
C:\Users\maart\AppData\Local\JetBrains\Toolbox\apps\Rider\ch-0\191.5812.55\bin\runnerw64.exe
Hosting environment: Production
Content root path: C:\Users\maart\Desktop\AcmeCorp.Web\AcmeCorp.Web
Now listening on: http://127.0.0.1:9131
Application started. Press Ctrl+C to shut down.
[IIS EXPRESS] Request ended: http://localhost:25367/ with HTTP status 200.0
[IIS EXPRESS] Request ended: https://localhost:44301/ with HTTP status 200.0
[IIS EXPRESS] Request ended: https://localhost:44301/ with HTTP status 200.0
[IIS EXPRESS] Request ended: https://localhost:44301/ with HTTP status 200.0
[IIS EXPRESS] Request started: "GET" https://localhost:44301/css/site.css
[IIS EXPRESS] Request ended: https://localhost:44301/css/site.css with HTTP status 304.0
[IIS EXPRESS] Request started: "GET" https://localhost:44301/js/site.js?v=4q1jwFhaPaZgr8WAUSr
[IIS EXPRESS] Request ended: https://localhost:44301/js/site.js?v=4q1jwFhaPaZgr8WAUSrux6hAuh0
```

Send debug request

When this option is enabled, JetBrains Rider will send an initial request (with a method) to the target ASP.NET Core application. This ensures the application i debugger is attached, even when the browser is not started.

Additional IIS Express arguments

This field lets you pass additional arguments to the IIS Express host (for examp tracing).

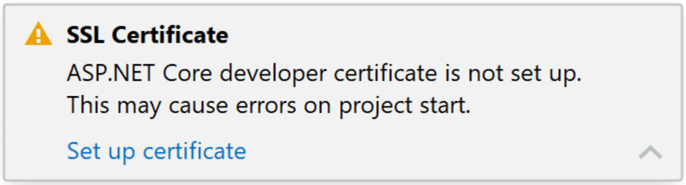
As with other launchSettings-based run configurations, you can specify the target framework and additional runtime arguments using this dialog, but some IIS Express-specific options should be configured directly in the **launchSettings.json** file.

Run and debug

Once everything is installed and configured, you can run your project Ctrl+F5 or set breakpoints (even in application startup, for example in `Configure`/`ConfigureServices` methods) and then debug your code Alt+F5.

Generate a self-signed SSL certificate

For launchSettings-based run configurations, JetBrains Rider will check whether an ASP.NET Core developer certificate or IIS Express self-signed certificate is set up on your machine. If not, you will see a notification which also let you generate one:



SSL Certificate

ASP.NET Core developer certificate is not set up.
This may cause errors on project start.

Setting up a certificate can be done from that warning, or with the **Set up certificate** action, which you can find using Find Action Ctrl+Shift+A.

For ASP.NET Core, JetBrains Rider will use the `dotnet dev-certs https` command to check whether the certificate has been installed. For IIS Express, it will check the http.sys certificate settings (similar to the `netsh http show sslcert` command line).
It also verifies the certificate is trusted by our system. This should make it easier to develop ASP.NET and ASP.NET Core web applications that make use of SSL on our developer machine.

Courtesy:https://www.jetbrains.com/help/rider/Running_IISExpress.html#generating-ssl-certificate

Modified: 2021.10.04.7.40.AM
Dököll Solutions, Inc.