Optional17: Deployment

Objective

Now you've built your website, you will want to deploy it.

ASP.Net Core supports a variety of deployment scenarios.

During development we've been using IIS Express when running from within Visual Studio but this is only intended for development. The built-in web server is very lightweight. For a production system, a fully-fledged webserver allows a much better degree of management of the application.The most common deployment scenarios for ASP.Net applications are probably deploying to either IIS or to an Azure App Service.

In this lab, you are going to deploy to IIS, and we'll take a look at some of the considerations that need to be taken into account during deployment.

This exercise will take around 30 minutes.

Referenced material

This exercise is based on material from the chapter "Deployment".

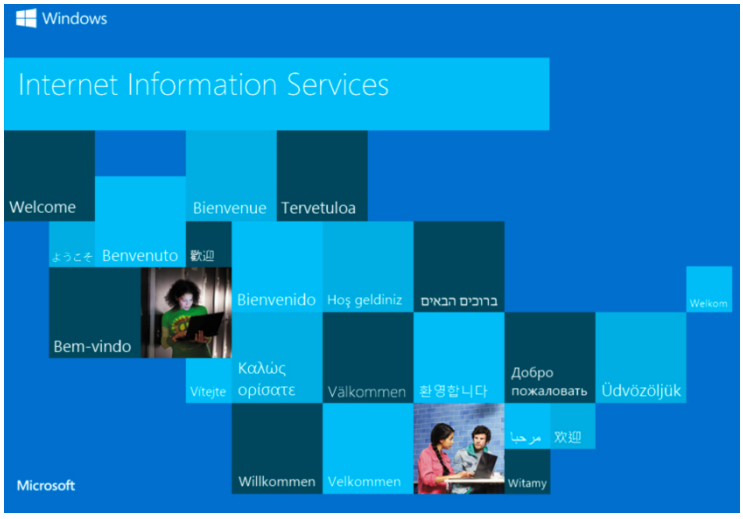
Preparing the Seed Data

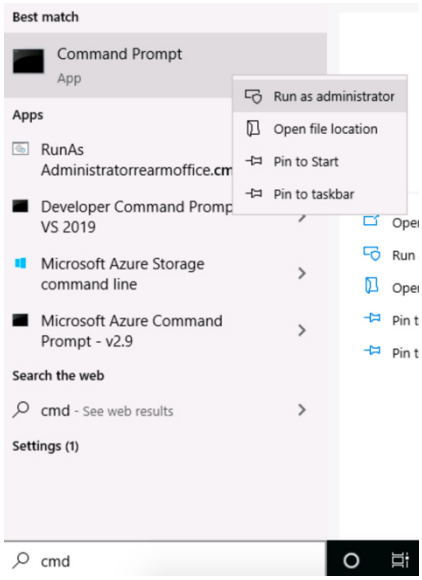
1	Open the 'Begin' solution in Visual Studio and compile (Shift Ctrl+B). The 'Begin' solution is identical to the 'End' solution from the previous lab.
2	Up until now, we've been using the Forum.Data database which was pre-created. Instead, we want to create the database as part of the deployment process and seed the initial set of forums, but not the threads or the posts. We will do this using Entity Framework's data seeding functionality.
3	So, open /EF/ForumDbContext. Find the method OnModelCreating(). The method already contains code which describes the foreign keys in our database. At the end of this code, add the following. You can change the number and name of the forums if you want: `Protected override void OnModelCreating(ModelBuilder modelBuilder)` `{` `modelBuilder.Entity<Thread>()` `.HasOne(t =>t.Forum)` `.WithMany(f =>f.Threads)` `.OnDelete(DeleteBehavior.Restrict);` `modelBuilder.Entity<Post>()` `.HasOne(p =>p.Thread)` `.WithMany(t =>t.Posts)` `.OnDelete(DeleteBehavior.Restrict);` `modelBuilder.Entity<Forum>()` `.HasData` `(`

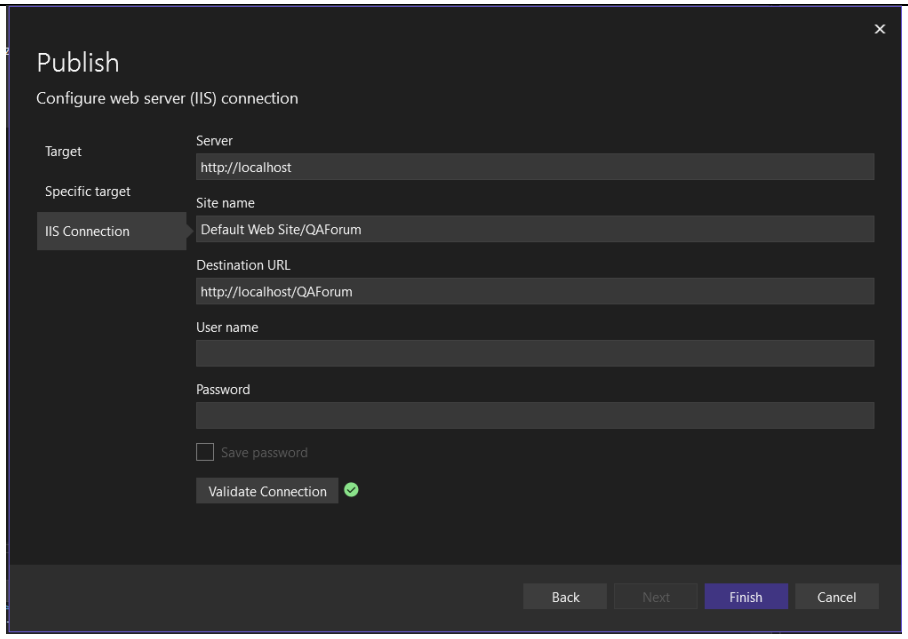
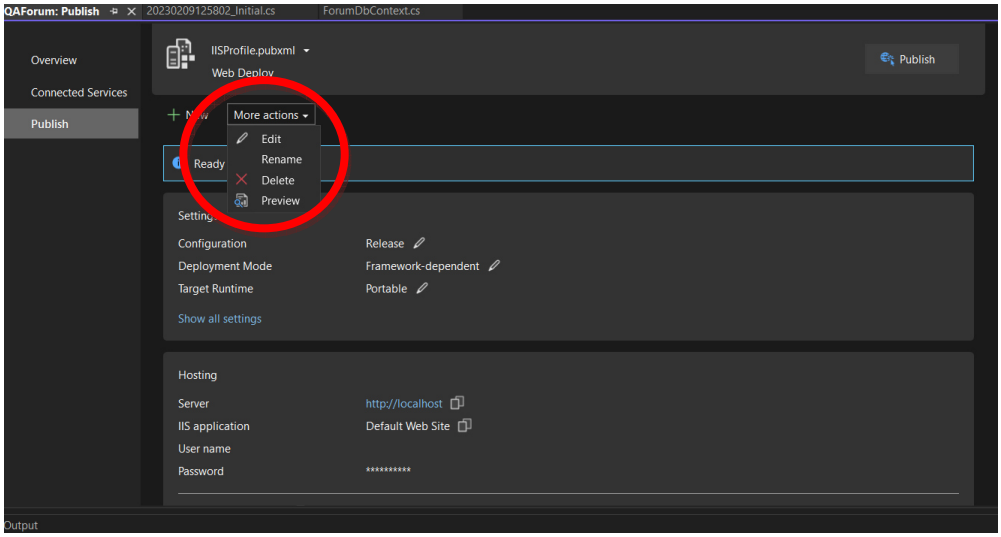
	`` new Forum { ForumId = 1, Title = "ASP.NET MVC" }, new Forum { ForumId = 2, Title = "ASP.NET AJAX" }, new Forum { ForumId = 3, Title = "ASP.NET Blazor" }, new Forum { ForumId = 4, Title = "jQuery" }, new Forum { ForumId = 5, Title = "Silverlight" }, new Forum { ForumId = 6, Title = "Visual Studio 2022" }, new Forum { ForumId = 7, Title = "WPF" }); } ``` Note that, when using the HasData() method, you have to provide the primary key, which is different to when you add data to the database in the normal way. HasData() is not intended to be used once the system is running – it is only to be used to seed data when deploying the application.
4	We will use the EF Core migration tools to generate the code which will be used to create the DB, create the tables and seed the Forum table. Go to Tools, Nuget Package Manager, Package Manager Console. In the package manager console, enter the following command: `add-migration Initial -context ForumDbContext` Once the migration has been added, the file containing the migration is open. Have a look through the file – you'll see that the forum data is included in there, as part of the migration.
5	Compile the app (Ctrl + Shift +B).

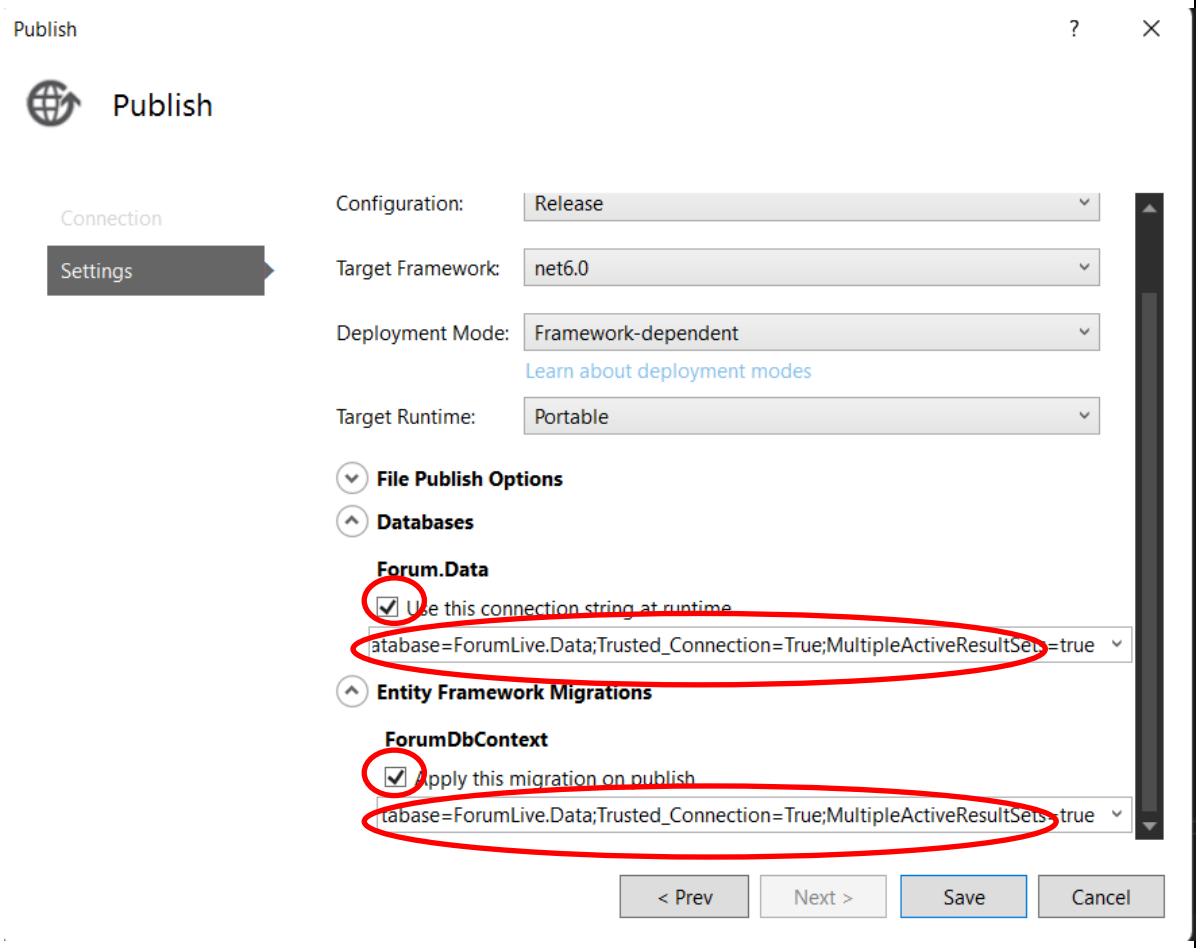
Deploying to IIS

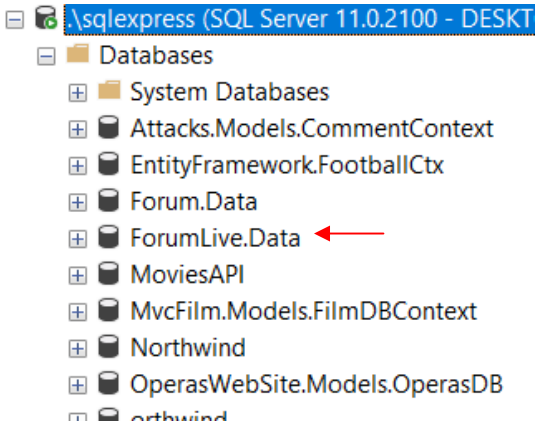
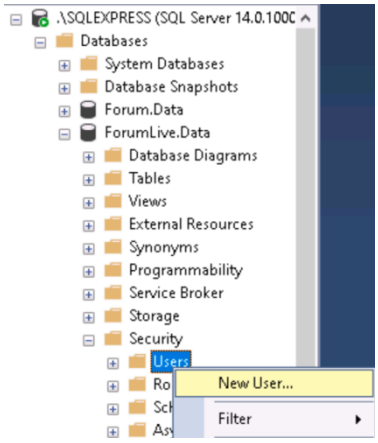
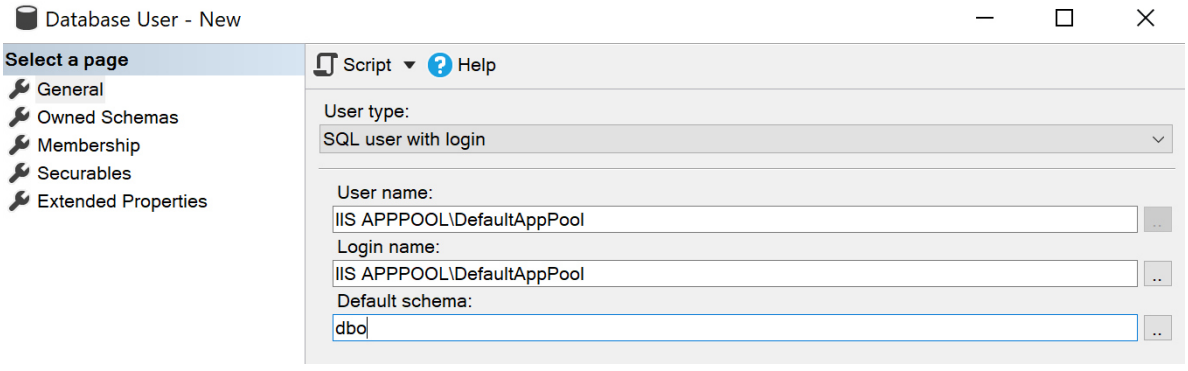
Next, we will deploy our application to IIS.

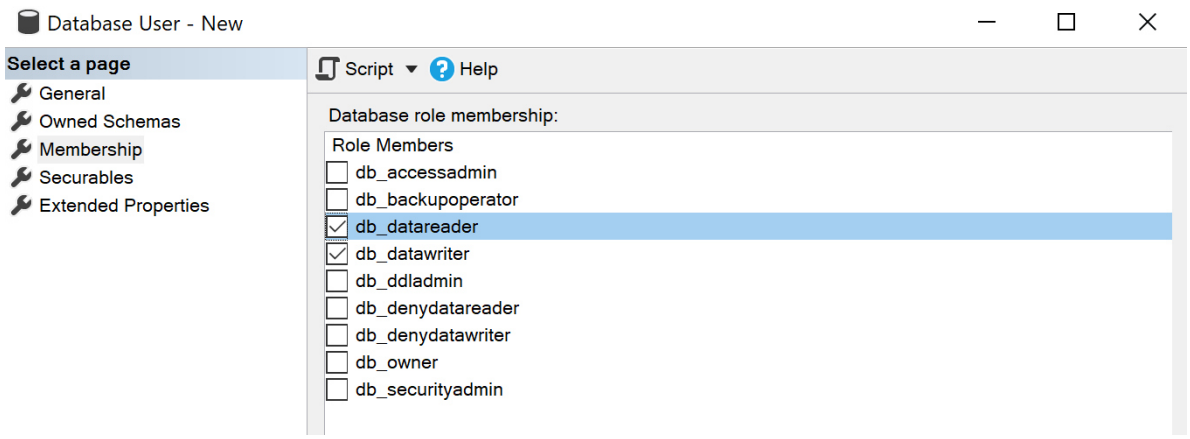
6	Ensure that IIS is installed on your computer. You can do this by opening a web browser and going to http://localhost. You should see a screen like the following:  If you see an error message instead of this screen, you will need to install IIS.

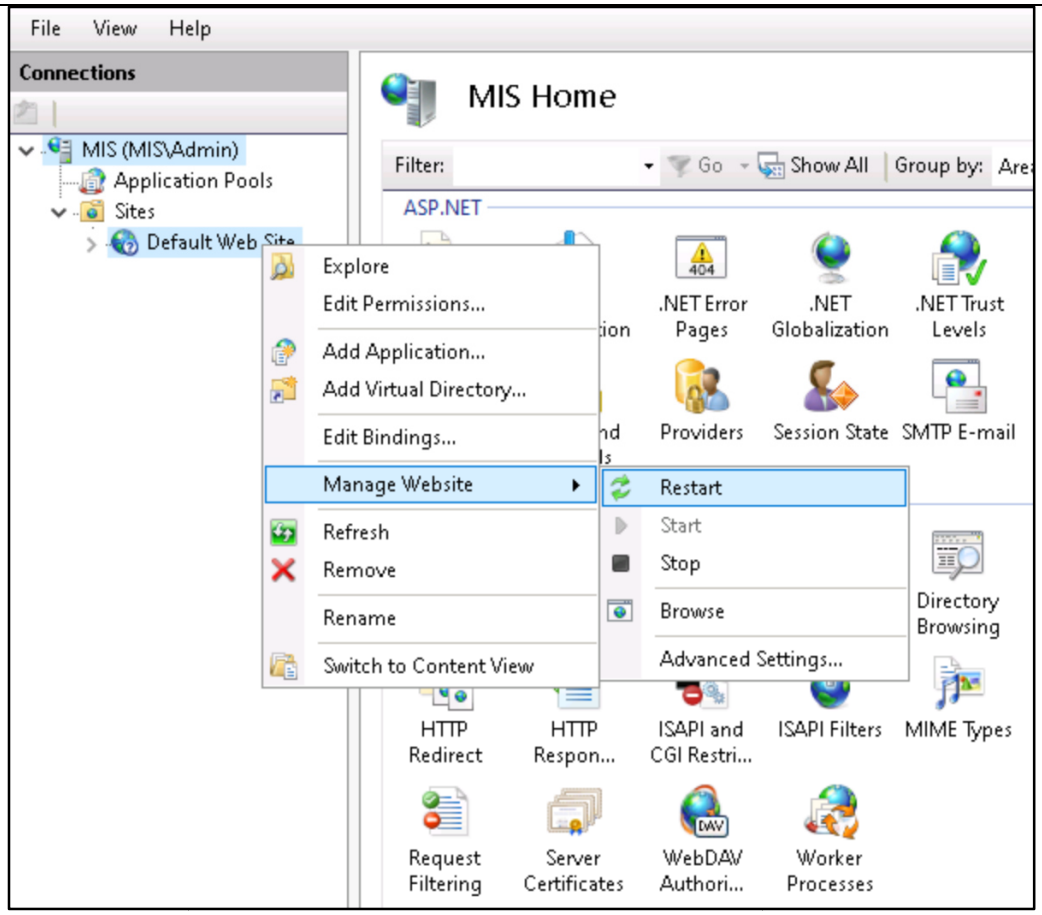
7	The Asp.net Core hosting bundle is required to get our app running on IIS. It will be already installed if you are use the image provided to do these labs. If not, you will be able to download an installer for it here:- https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-6.0.13-windows-hosting-bundle-installer Download the version to match the version of .net you used to complete the labs. The purpose of this bundle is to allow IIS to host ASP.Net Core applications.
8	After installing the hosting bundle, IIS will need to be restarted. Use the following steps to restart it: • Click on the start menu, and type "cmd" • The "Command Prompt" app will be shown. Right-click on the icon for Command Prompt, and choose Run As Administrator: • In the command window that opens, type the following two commands: `net stop was /y` `net start w3svc` 
9	Now that we know that IIS is installed and ready for our .Net Core application, we need to prepare to deploy. This needs to be done as an administrator. So, close down Visual Studio. Run Visual Studio 2022 but this time as an Administrator (a right click option). Once Visual Studio starts, open the QAForum project (it will probably be at the top of your Recents list, but otherwise you will need to navigate to it. Make sure you open the version from this lab, not from a previous lab!)
10	In the solution explorer, right-click on the QAForum project, and choose Publish.
11	At the next screen, you are asked to pick a publish target. Choose "Web Server (IIS)", then click Next.
12	Ensure that the publish method is set to Web Deploy and click Next. Then enter the following details: • Server: http://localhost • Site name: Default Web Site/QAForum • Destination URL: http://localhost/QAForum Click the Validate Connection button to make sure you've entered these details correctly:

	
13	Click Finish then click Close in the Publish profile creation progress dialogue window In the QAFurum: Publish screen choose Edit from the More options dropdown.  Wait for “Discovering Data Contexts” to complete if necessary. Then, expand the “Databases” option, and the “Entity Framework Migrations” option
14	Under the “Databases” option, you will see the connection string that is in our program’s configuration file. Tick the checkbox. This tells Visual Studio that we’d like the live version of our program to use a different connection string. Change the first part of the Database value from Forum to ForumLive, so that it is now ForumLive.Data
15	Now, under Entity Framework Migrations: • Tick the box to say that you want to apply the database migration. (The migration will create the database, create the tables, and add the seed

	data.) • Copy the connection string from “Forum.Data” in the Databases section to “ForumDbContext” in the Migrations section
16	This image shows what the screen should look like know, and indicates the areas you should have changed: 
17	Click Save to save this publish profile. You can now re-use the profile every time you need to publish your application.
18	Click Publish. Your program will be compiled, packaged up for publishing, and then sent to the web server. Then, the database migrations will be applied. Once all of this is complete, a web browser will open on your website’s home page… but you will see an error.
19	The reason for the error is that the web server runs under a different user profile to the one we’ve been working with throughout the course, and that user profile does not have permission to access the databases.

	To fix this, open SQL Server Management Studio.
20	Check in SQL Server Management Studio that two new “live” databases have been created: 
21	Expand ForumLive.Data, then expand Security.  Right-click on Users, and select New User. In the New User dialog, select a user type of “SQL user with login”. Then, enter the following for both the user name and the login name. It must be entered accurately, because this is the name of the account which IIS is using: IIS APPPOOL\DefaultAppPool For the default schema, enter: dbo The screen should look like this: 
22	Click on the Membership section of the new user screen. Put ticks in the boxes labelled: • db_datareader • db_datawriter

	The screen should look like this: 
23	Click the Ok button to save the user details. You will only need to complete the last few steps the first time you deploy, when a new database is created. Once the database is created and the permissions have been set, they will remain set when you deploy again, even if your application includes new migrations that need to be applied.
24	IIS now needs to restart the application, otherwise it remains in its error state. Again, this will only need to be done the first time you deploy the application. Click on the Start menu, and type “IIS”. Of the options presented, you should be able to see “Internet Information Services (IIS) Manager”, which you should now click on. In the left-hand side of the application, expand the computer’s name. Then expand the “Sites” section. Right click on Default Web Site, and choose Manage Website, Restart:

	
25	Now, refresh your web browser. (If you closed the web browser, open it again and visit http://localhost/QAForum) Everything should be working! Look at the Forums list – you should see the forums that you seeded using Entity Framework. But there will be no threads or posts until you add some.

Congratulations, you now know how to deploy your software! Let's take a brief look at what happened to our settings file when we deployed the application.

Understanding Settings Transformations

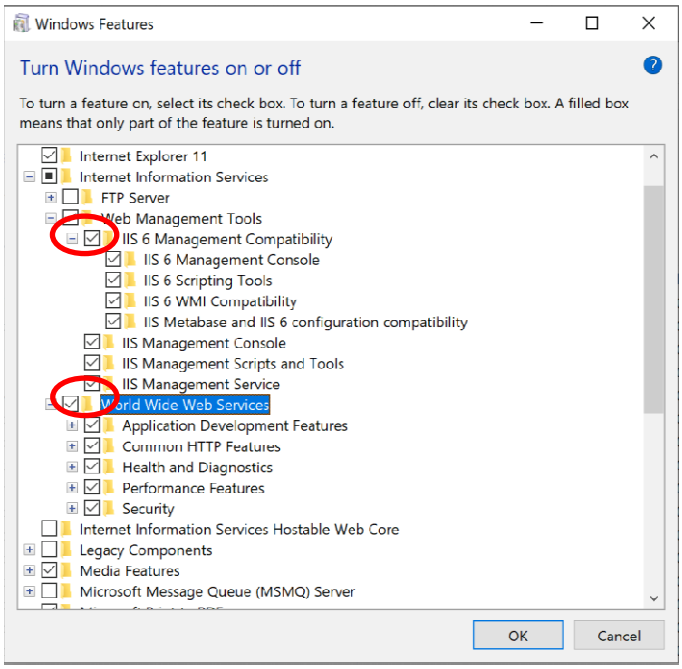
26	One last thing to do. Have a look in C:\Inetpub\wwwroot\QAForum. This is the folder where your website files are located. Find the files that start with "appsettings…" There is a file called appsettings.production.json, which was not in our original project. Use whatever text editor you like to take a look in this file. Any settings in this file will override the standard settings, so long as the ASPNETCORE_ENVIRONMENT variable is set to "production". In here, you will find the new connection strings. This file was created as part of the deployment of your project.
27	While we're here, have a look at one more file: web.config. This file was not in our original project at all, but has been added as part of the deployment. It contains IIS configuration options. Many options can be set graphically through the IIS Manager, but further options

	can be set in this file if required.

That's the end of the course – well done!

Installing IIS

You should find that IIS is installed on your lab computer. But in case it isn't, or if you want to know how to install IIS on your own computer, we've provided instructions for installing it below:

28	To install IIS, click the Start menu, and type "Control Panel". Open the control panel. Click "Programs and Features"
29	Click "Turn Windows Features On or Off" From the options that are shown, click the + sign next to "Internet Information Services
30	There will be three sections available, each with their own sub-sections. Go into the sub-sections for Web Management Tools and for World Wide Web Services, and turn on everything in both sub-sections. By the time you finish, your screen should look like this: 
31	Click Ok, and wait for everything to install. You may need to reboot your computer.