# Booking Block: Deployment into Development and Production Environments

This guide shows you have to take the code for Booking Block and deploy it into a development environment (such a Visual Studio) or into a live production environment (such as Azure).

## What is Needed

* Microsoft Visual Studio 2015
* SQL Server Express (for development)
* Backend code (Web API/SQL Server) from github
* Frontend code (HTML/AngularJs) from Cloud9.
* Azure account (for Production).

Microsoft Visual Studio 2015 (Community Edition) can be downloaded from <https://www.visualstudio.com/>

SQL Server Express 2014 can be downloaded from <https://www.microsoft.com/en-gb/server-cloud/products/sql-server-editions/sql-server-express.aspx>

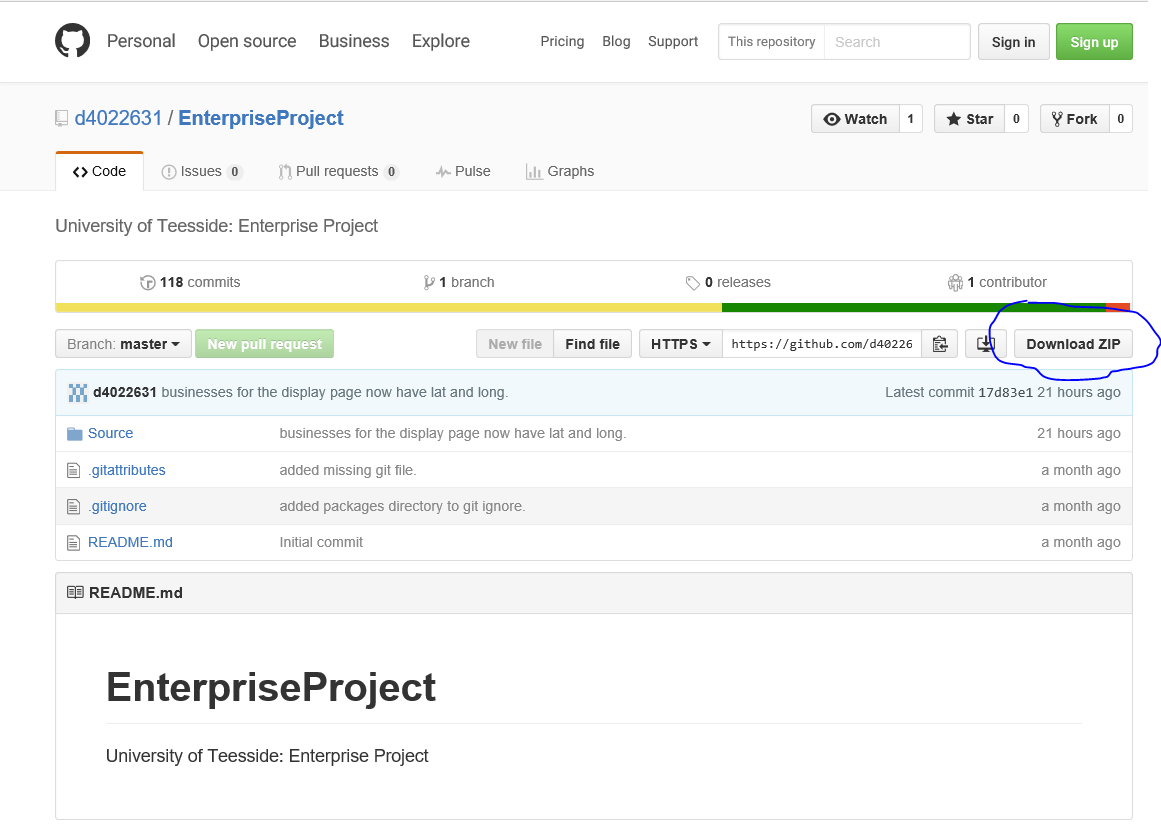
## Getting the code

### Web API from Github

The github repository for Booking Block can be found at:

<https://github.com/d4022631/EnterpriseProject>

Click the “Download Zip” button to get the latest version of the code.

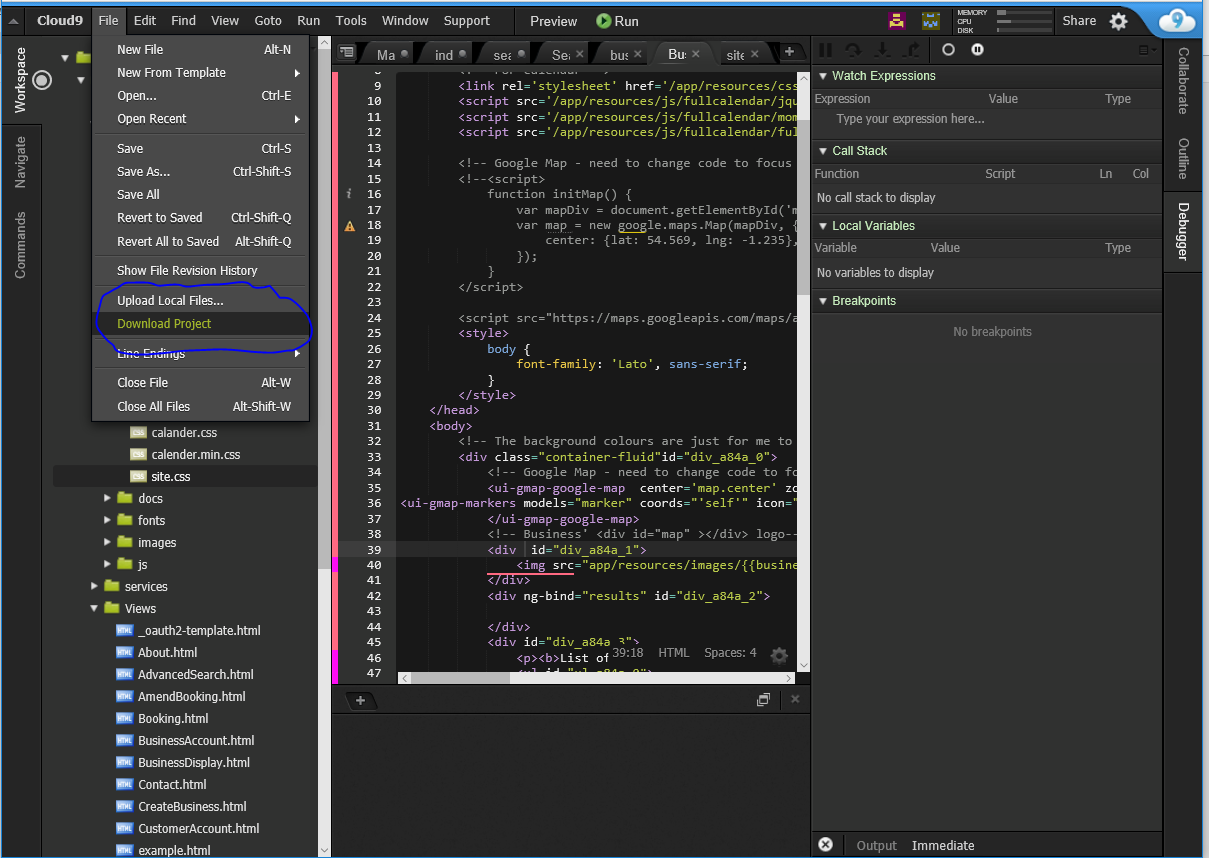


### Front End from Cloud9

The cloud9 workspace can be found at:

<https://ide.c9.io/d4022631/enterpriseproject>

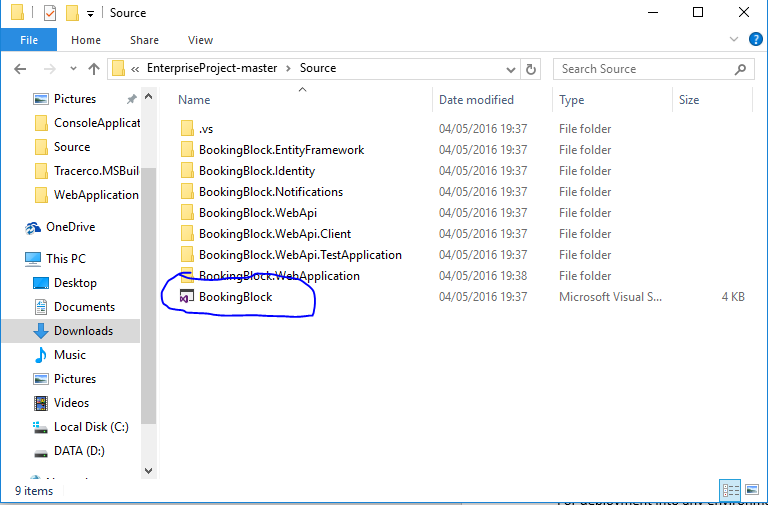
From the “File” menu select “Download Project”



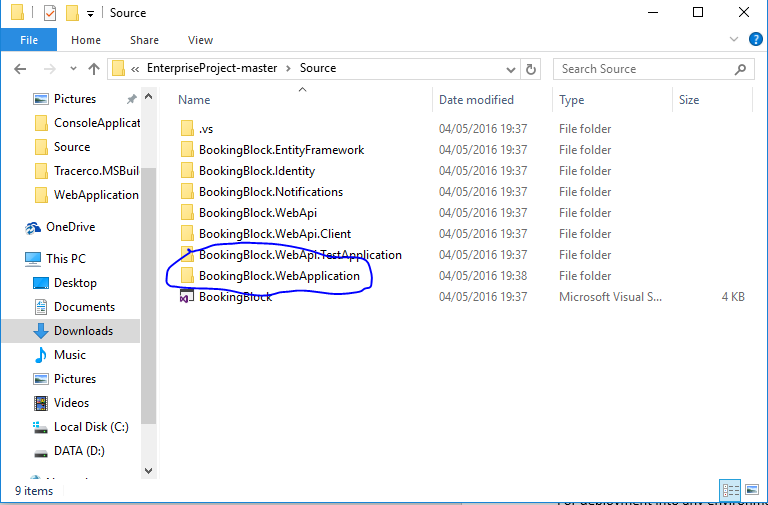
## Combining the Code

For deployment into any environment the two zip files need combining into one project.

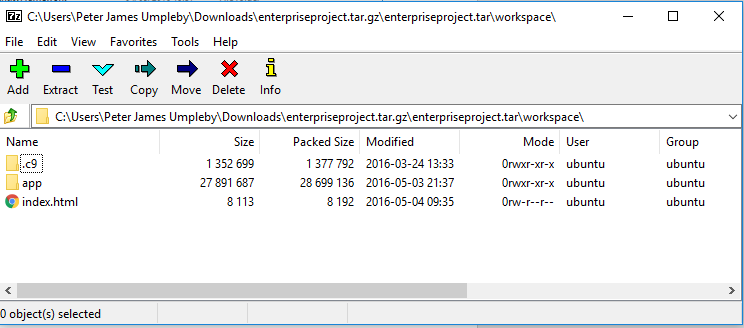
First extract the “EnterpriseProject-master.zip” from Github. After extracting the files you can find the Visual Studio solution in the sources directory.



The folder named “BookingBlock.WebApplication” is the root of the website when it is deployed, it is into here we need to put the files for the front end from cloud9.

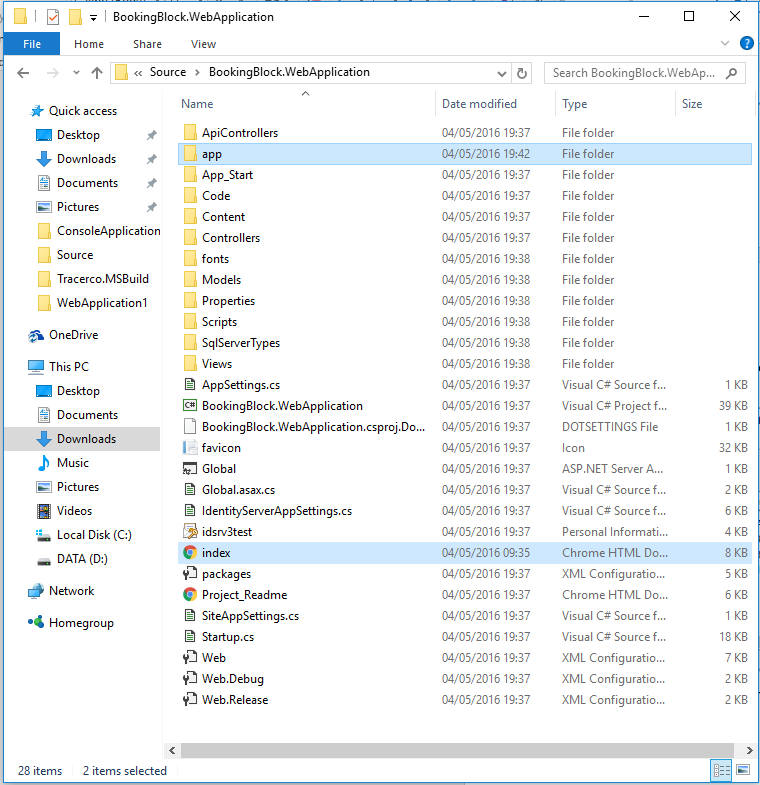


The “enterpriseproject.tar” from cloud9 contains the front end files, inside of the tar archive there is a workspace folder. It is the files in this folder we require.

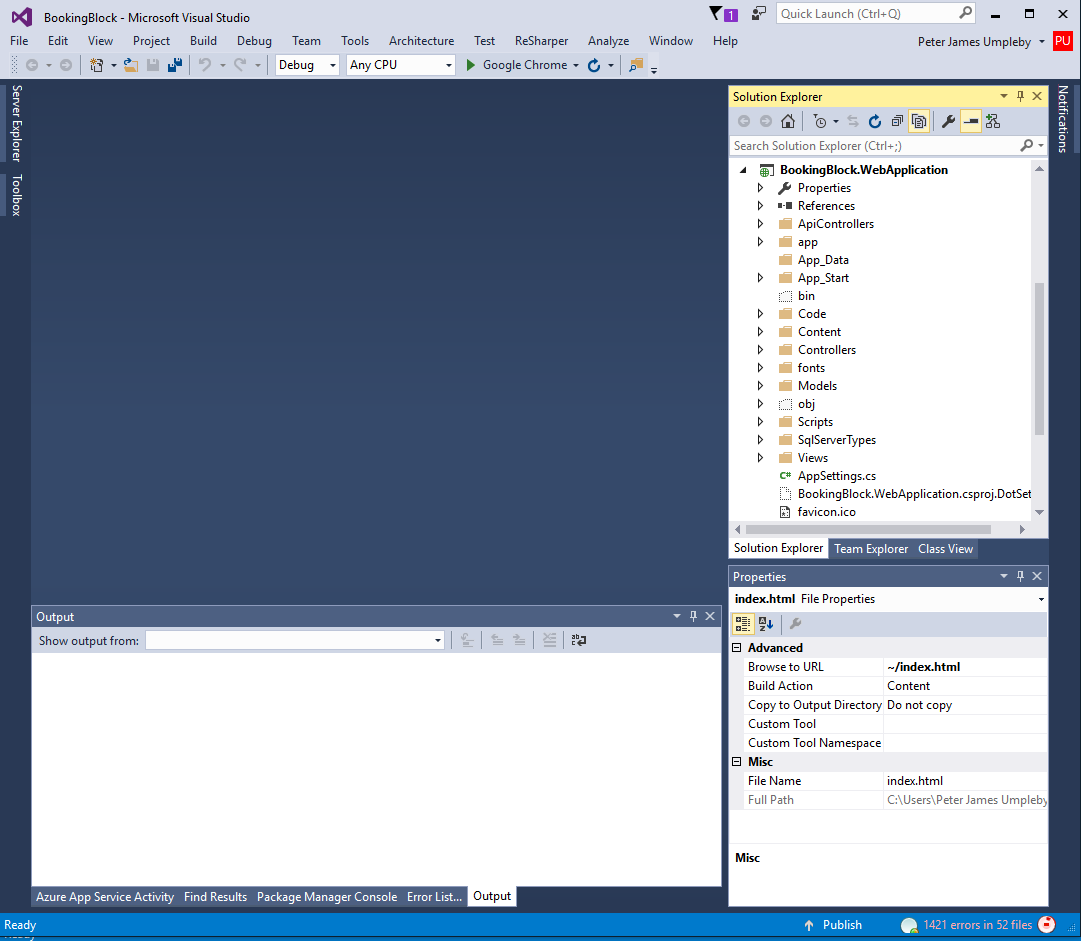


These files want extracting into the site root folder “BookingBlock.WebApplication”.

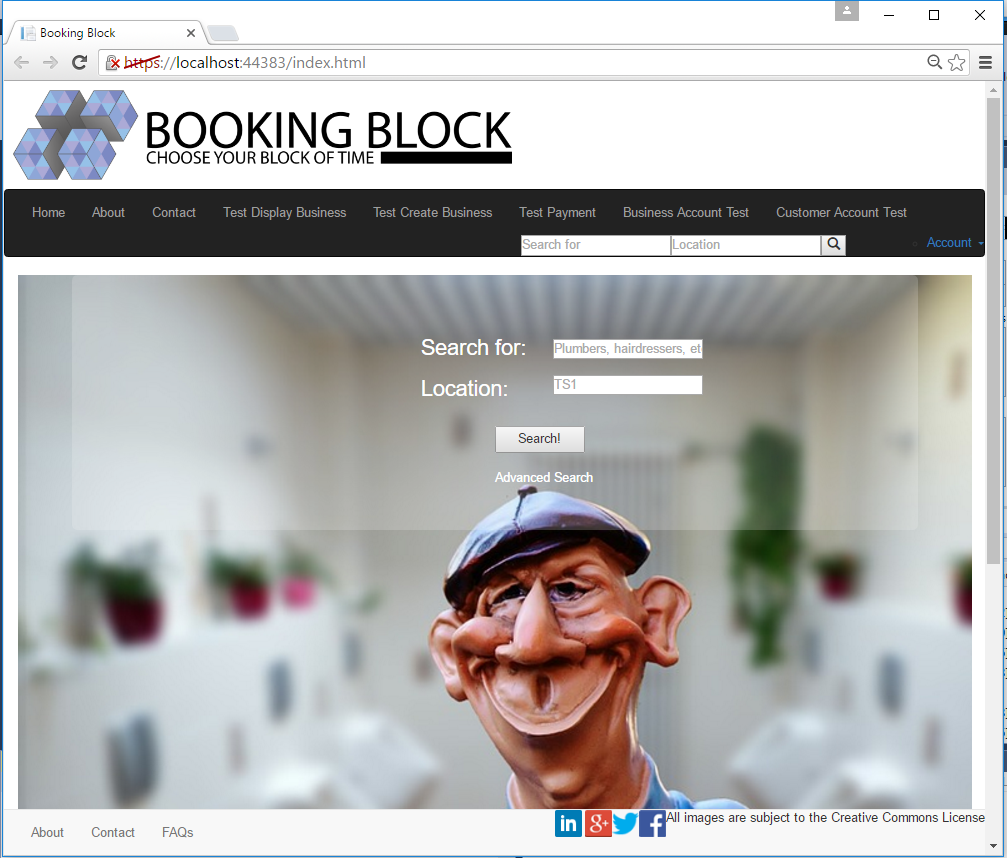
The “BookingBlock.WebApplication” should now contain the files for both the front end and the back end.



The “BookingBlock” solution can now be opened in Visual Studio.



With the “BookingBlock.WebApplication” project set as the start-up project you should be able to run the site by pressing the debug button (F5).



By default the system uses a SQL LocalDB for the database assuming SQL Express is configured correctly no additional changes are needed to be made.

By default all new installations have a default admin account created.

Username: [webmaster@bookingblock.azurewebsites.net](mailto:webmaster@bookingblock.azurewebsites.net)

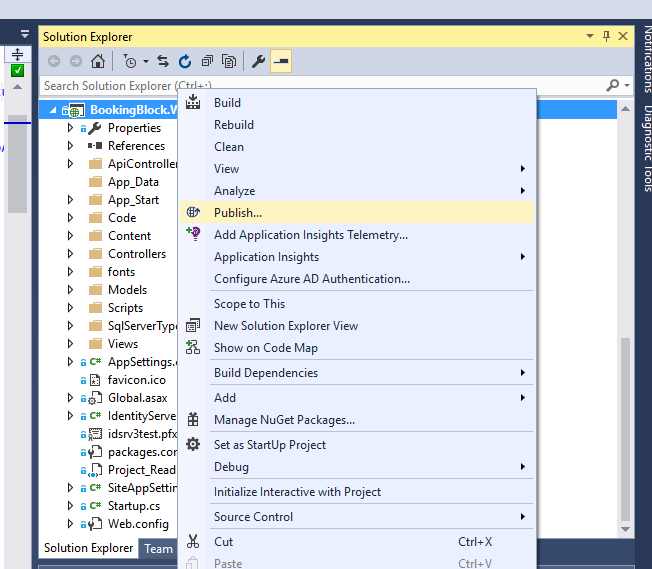
Password: Enterprise2016!

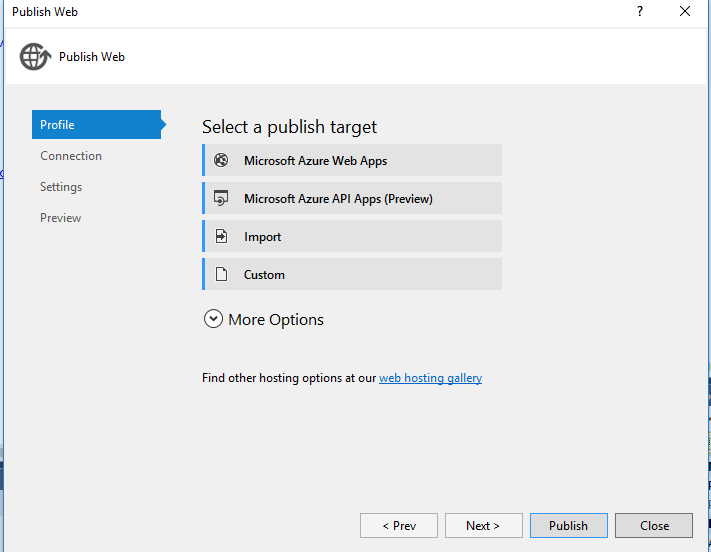
\*n.b. when running locally single sign-on from third-party sites (Facebook, Microsoft, Twitter, etc..) will not operate correctly, so local accounts are required for testing.

This setup can be used for development for the next section looks at deploying the application into the Azure cloud.

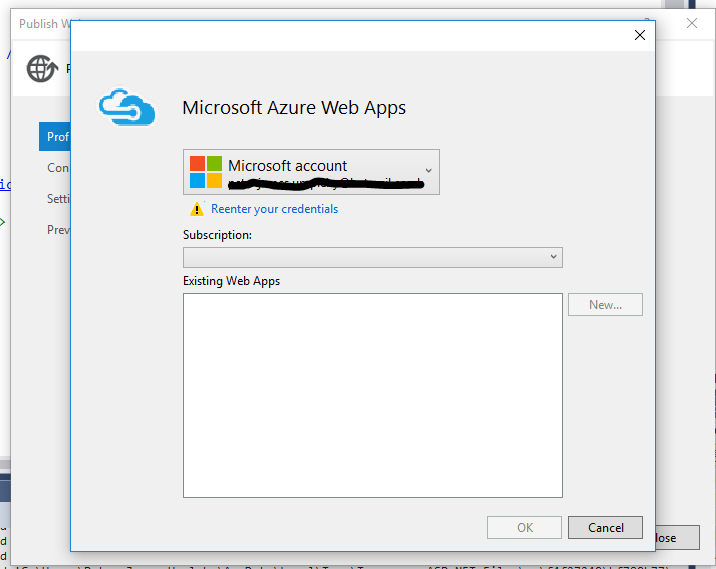
## Production Deployment

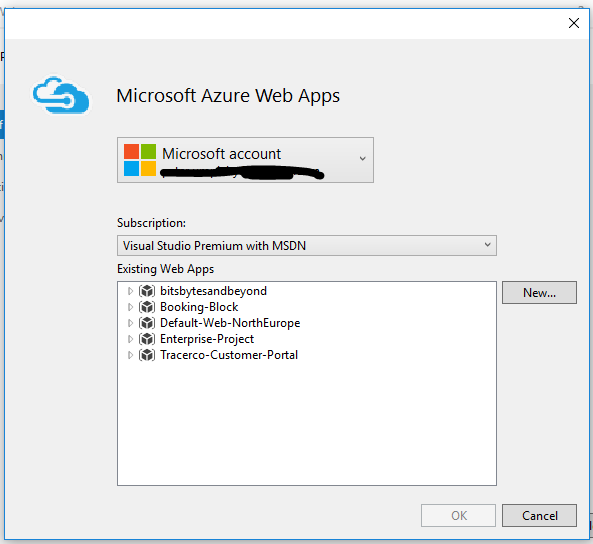
From Visual Studio right-click on the BookingBlock.WebApplication project and select publish.



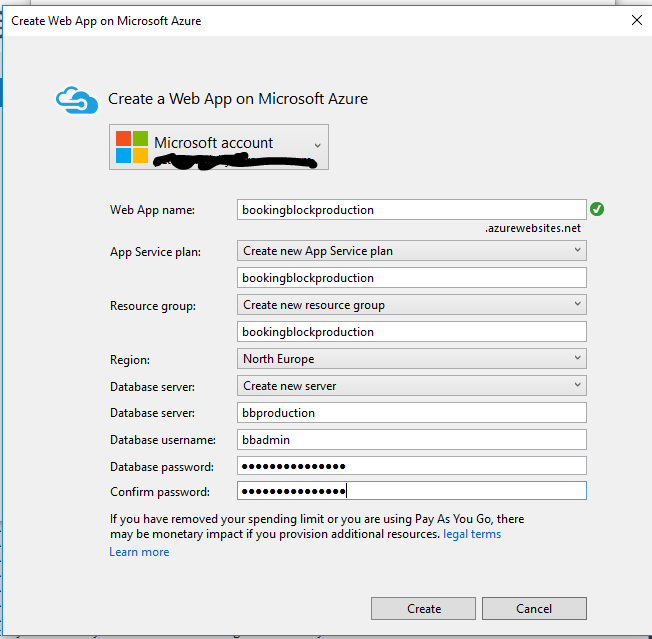


Next we need to select “Microsoft Azure Web Apps” as the target. On the next screen you will need to sign into your azure account.

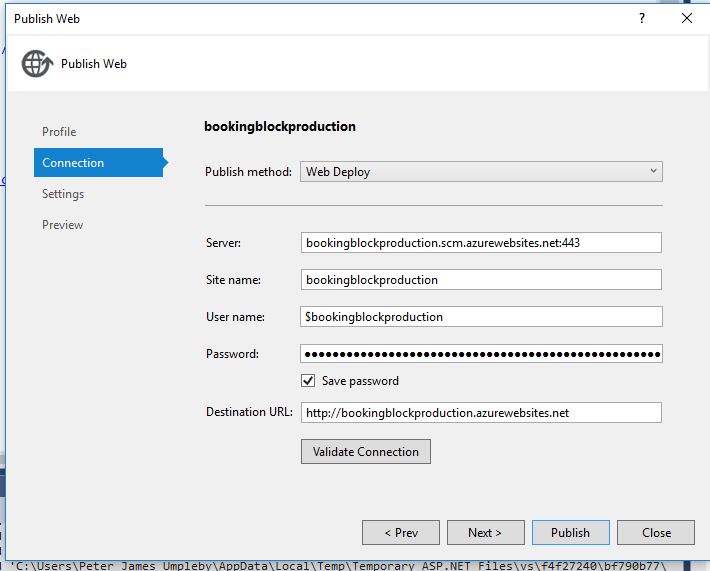




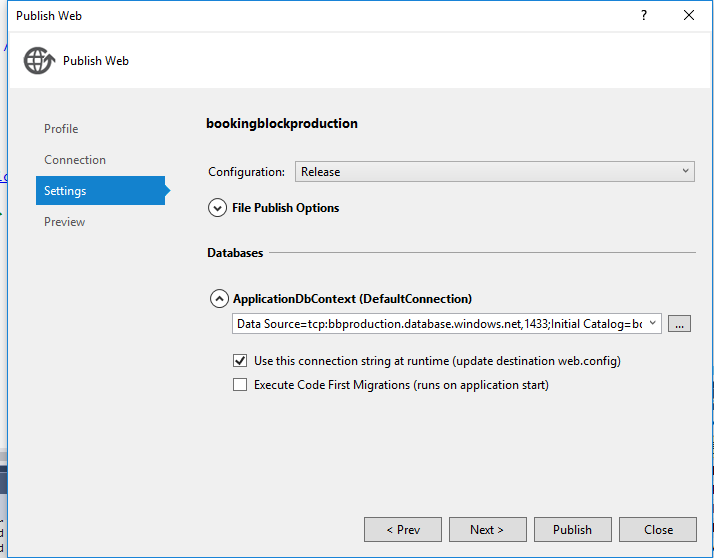
Now select “New…” and setup your new azure web app.

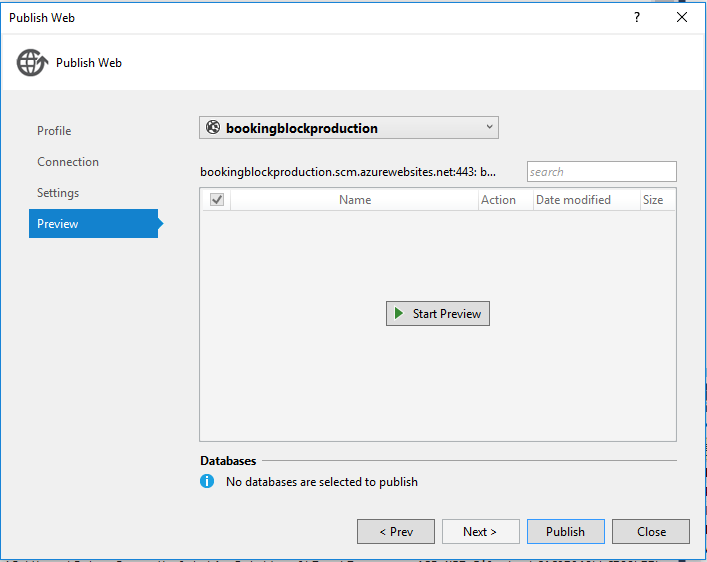


Once complete your site will be ready to deploy

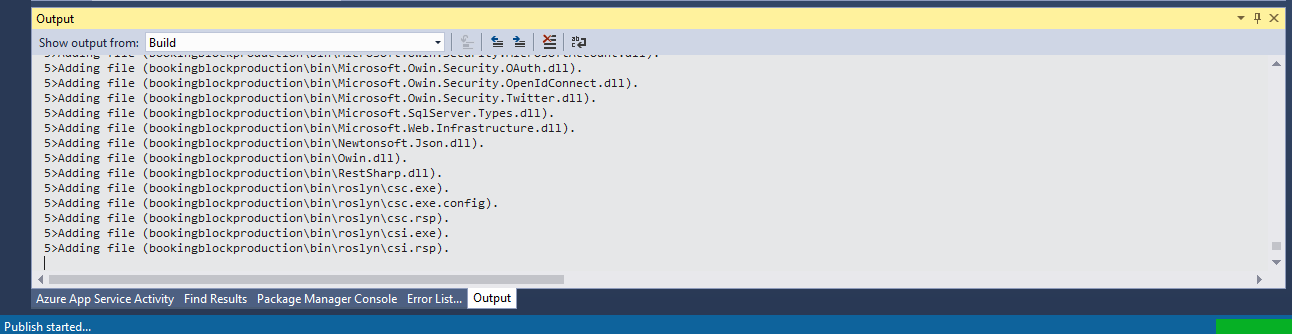


**On the next screen make sure that “use this connection string at runtime” is checked. This replaces the LocalDB connection string (which won’t work on azure, with the connection string for the Azure hosted SQL server).**



On the next screen press “Publish”  


The site will now be published to Azure.



The site should now be available on the URL you chosen for your web app. E.g. <http://bookingblockproduction.azurewebsites.net/>

