Publishing and MVC to Azure with DB Access

There is little literature about how to do this, so here are the steps:

On the Azure Portal we create a Database:

<https://manage.windowsazure.com/@rnmisrahigmail.onmicrosoft.com#Workspaces/SqlAzureExtension/Databases>

The best link is this one:

<https://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-deploy-aspnet-mvc-app-membership-oauth-sql-database/>

|  |  |
| --- | --- |
|  | We use Quick Create and enter a Database name. The subscription is whatever was set before and we use a server that was defined previously. The 10 character server name is important for the following steps.  g2ojj5kj5y |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  | If we create an MVC project with the MVC pattern.  We’ll get also a DB that contains the Authorization infrastructure to register and log in users. |

This infrastructure is a set of tables that point to a DB where the connection string is automatically named DefaultConnection and has this form:

|  |
| --- |
| <connectionStrings> <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\aspnet-WebApplication15-20160217124855.mdf;Initial Catalog=aspnet-WebApplication15-20160217124855;Integrated Security=True" providerName="System.Data.SqlClient" />  </connectionStrings> |

As you can see, the DB points to a local database. This means that if we run the just created app, we’ll be able to register users and by doing so, populate the database.

|  |
| --- |
| If you open you MS’s DBMS, you’ll see it |
|  |
| Note that the name follows the connectionstring DefaultConnection, which is: <namespace>-<project name>-<year><month><day><hour><minute><second>. |

If we want to publish this app in Azure, we follow these steps:

Note: To have access to publishing in azure, we have to set our IP in Azure’s firewall. The “old” Azure website is easier: Select the SQL DB and the click on [Set up Windows Azure firewall rules for this IP address](https://manage.windowsazure.com/@rnmisrahigmail.onmicrosoft.com).

<https://azure.microsoft.com/en-gb/documentation/articles/sql-database-configure-firewall-settings/>

Right-click on the project and select Publish:

|  |  |
| --- | --- |
|  | Select Microsoft Azue App Service |
|  | Subscription will show whatever subscription you have.  Select Resource Group.  In the Serach field you can select a previously defined WebApp hosted, or you can create a new one: Select New |
|  | Fill the fields as you please and select OK. |
|  | Back at the creation of your Web App in Azure, press the Create button.  At this moment, your Web App will be created. You can even see it in your Azure Portal as we’ll see later on. |
|  | The next step of the wizard is shown with important parameters:  The Destination URL and the Server, which is needed by the ConnectionString.  Copy the Server String. We’ll need it in the next step of the Wizard.  At this point I recommend to Validate Connection.  Select Next. |
|  | It may take a few seconds for this screen to refresh.  On the ApplicationDbContext we need to enter the string will replace our DefaultConnection in our local app.  This can be done by selecting the …button to the right of ApplicationDbContext |
|  | Where do we get the Server name?  We got the 10-character string when we created our web app in Azure.  The following screens shows the Azure screen where we can see it clearly: |
|  | We select the FirstDb we created in the beginning… |
|  | The string shown:  g2ojj5kj5y.database.windows.net,1433  Is the server name, which we need to create the ConnectionString. |
|  | After entering our Server Name and pressing OK we go back to the publishing screen with the BookContext string, which will have this form:  Data Source = g2ojj5kj5y.database.windows.net, 1433; Initial Catalog=FirstDb; User ID=<your Azure user name>; Password=<your Azure password> |