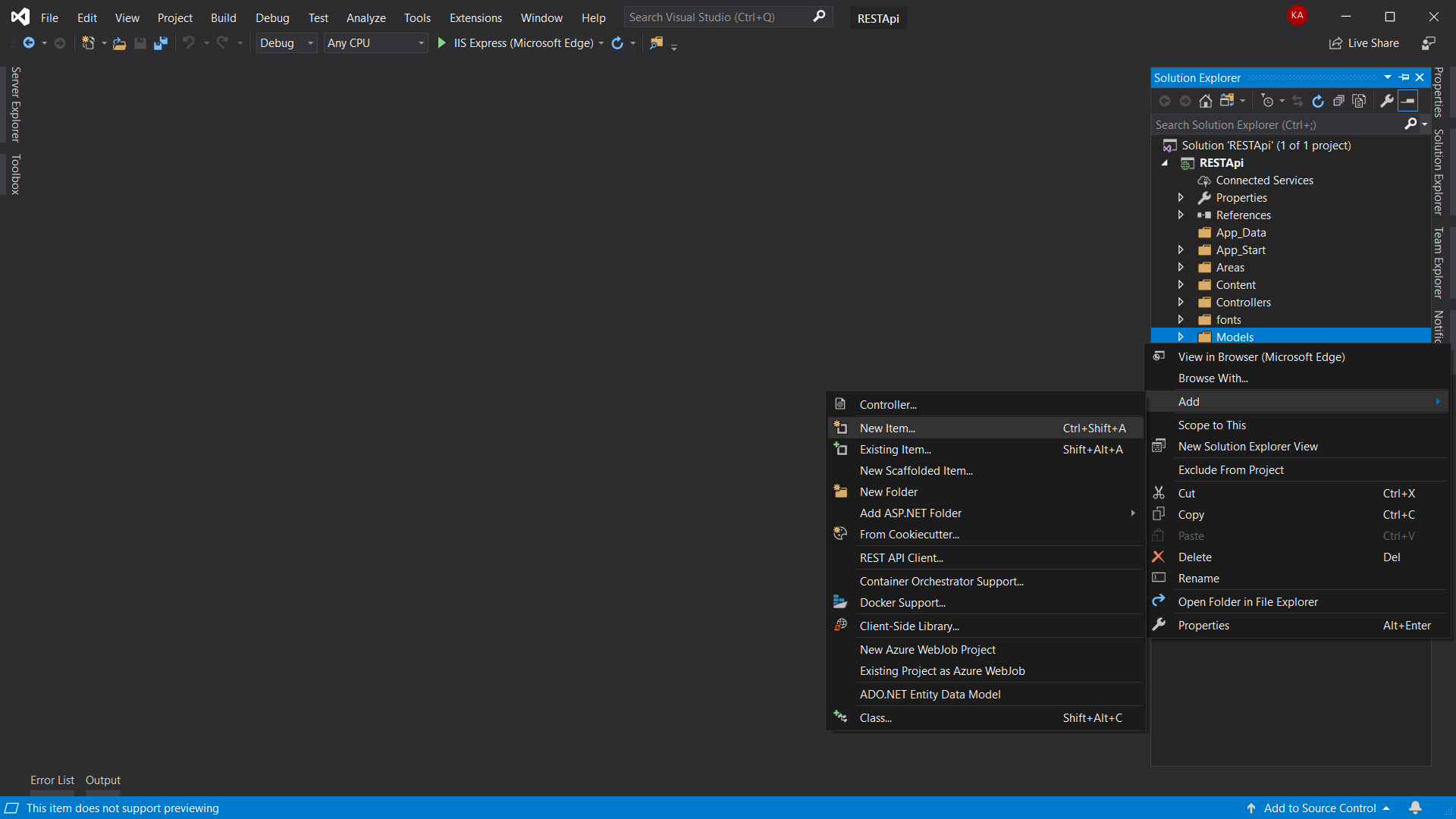
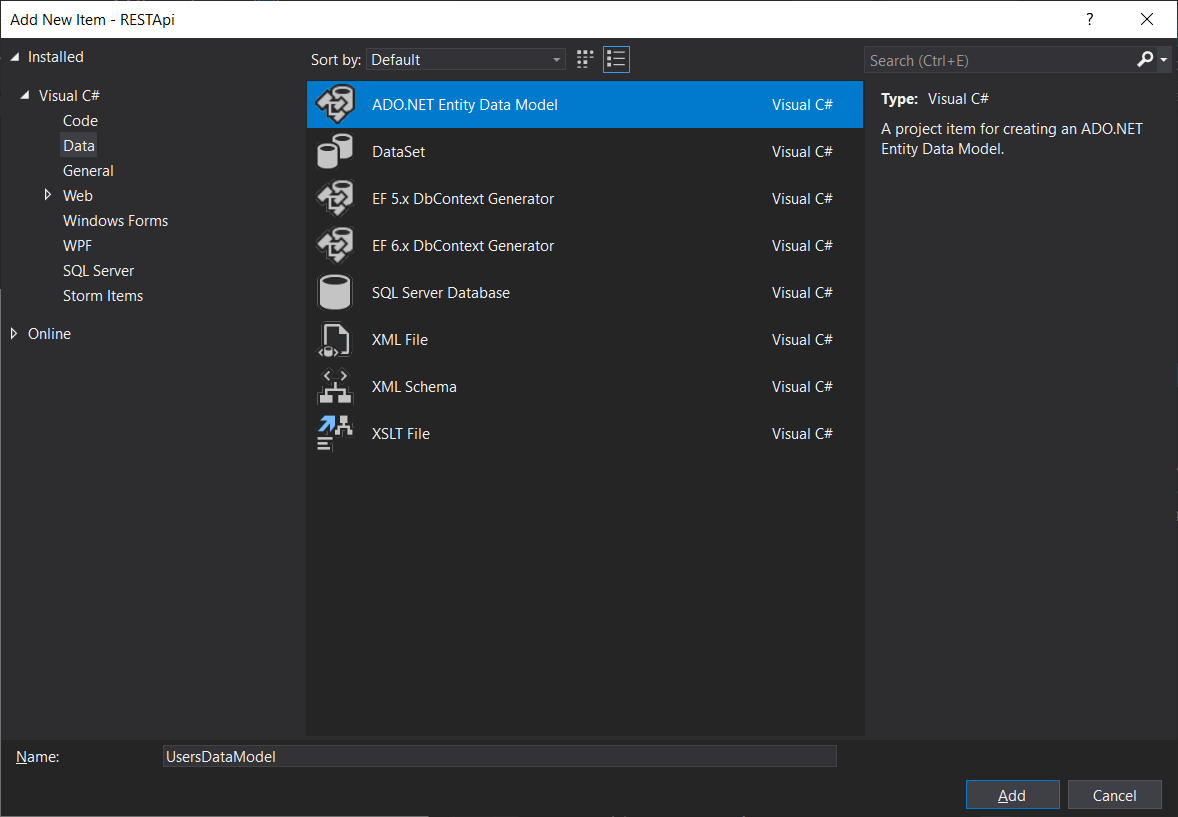
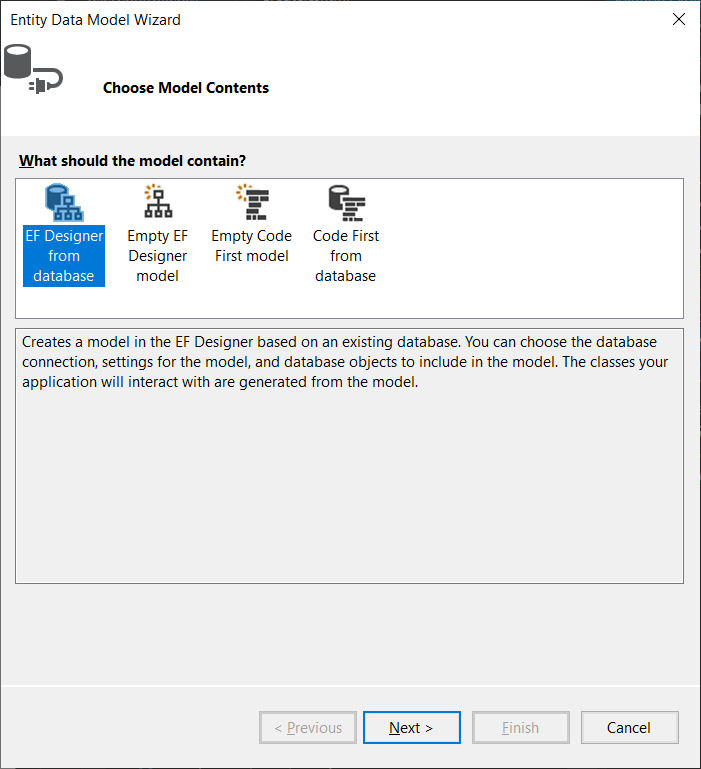
Setup Entity Data Model from Local Database in Visual Studios



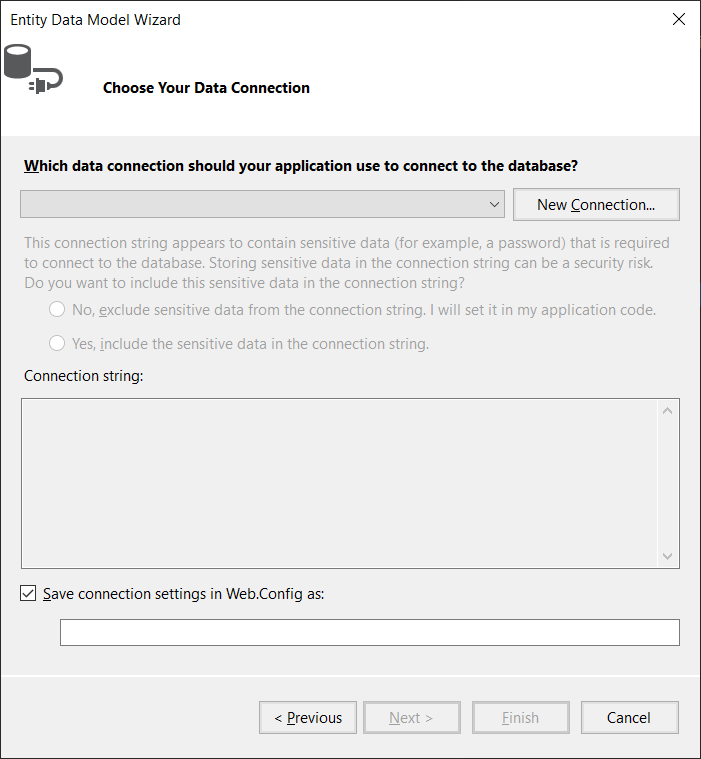
Right click Models and select Add 🡪 New Item.



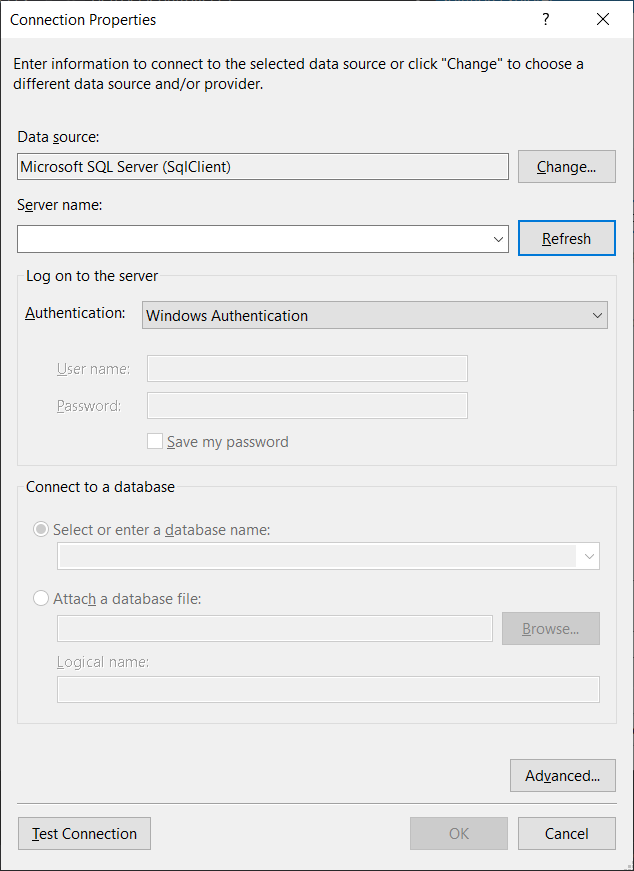
From the side bar select Visual C# 🡪 Data and highlight the option ADO.NET Entity Data Model, name the file and click the add button at the bottom right.



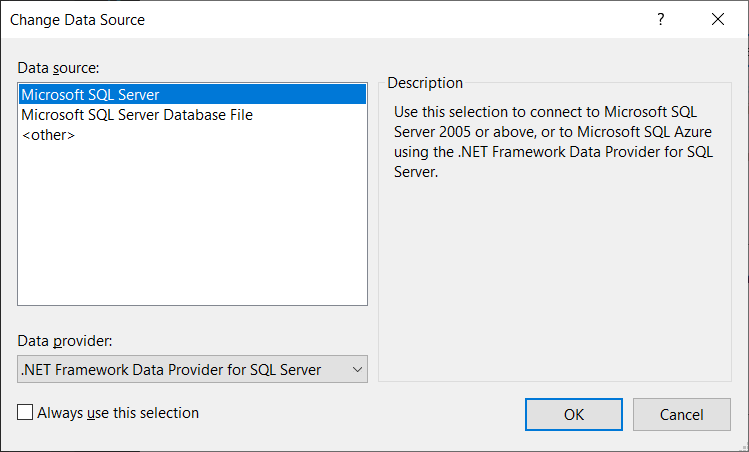
You should now be prompted to choose model contents in an Entity Data Model Wizard window. Select EF Designer from database and click next.



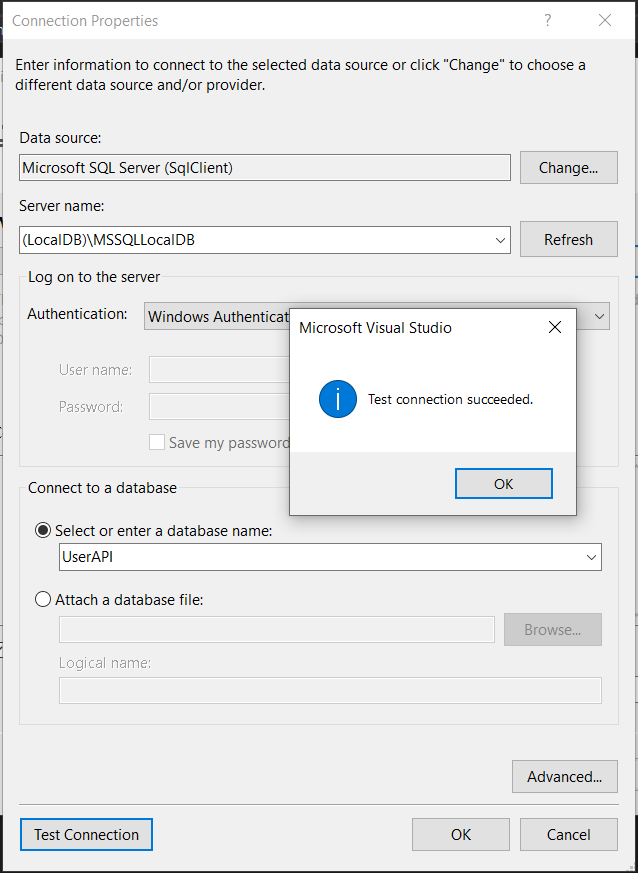
This should allow you to choose your data connection. Click on the New Connections button.



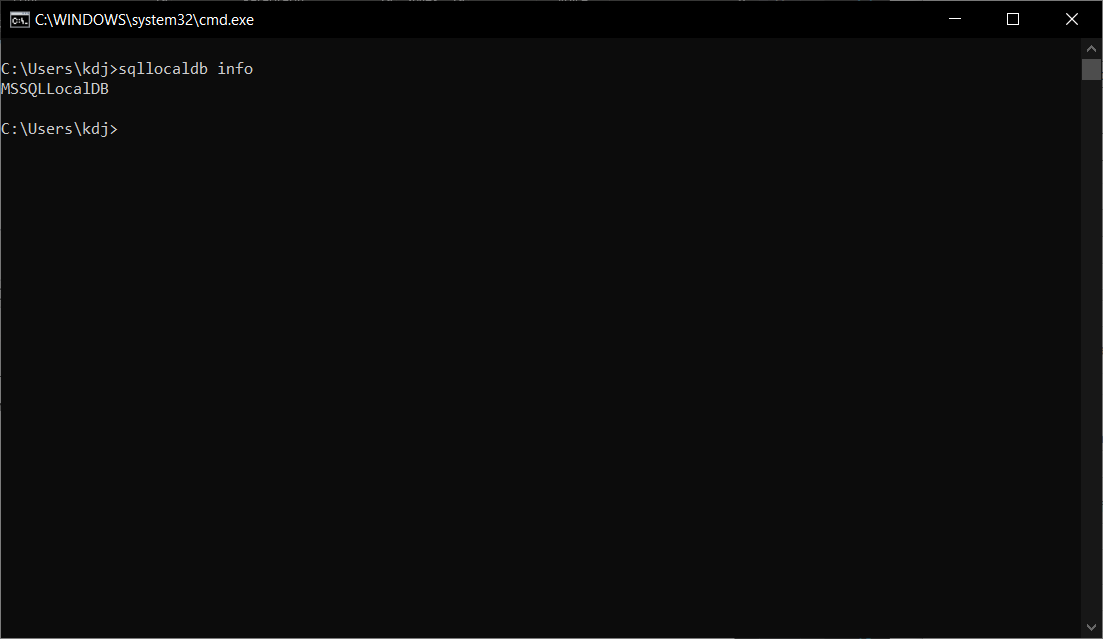
This will show the Connection Properties. In the Data source input select Microsoft SQL Server (SqlClient) via the Change button.



The change button will display the above windows. Now click OK.

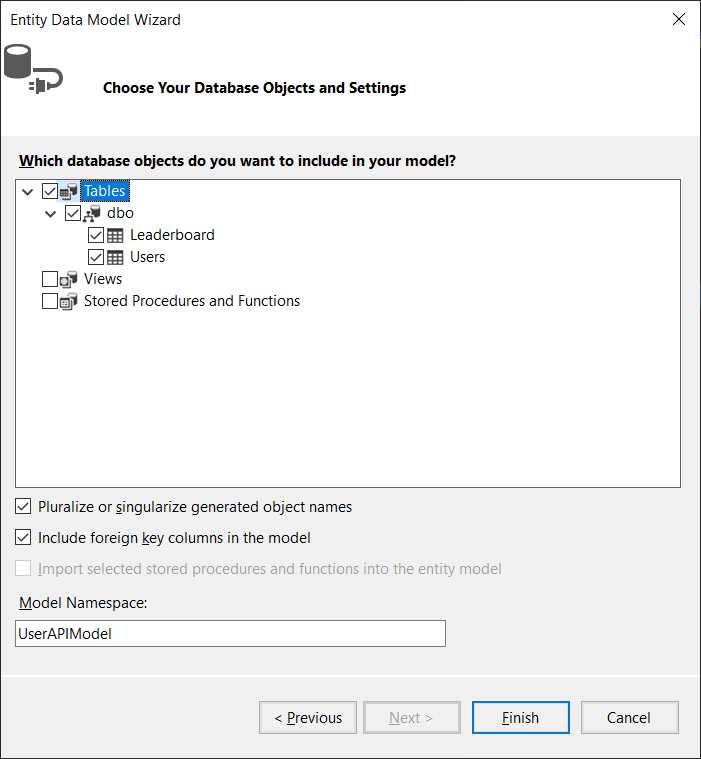


Now you should be returned to the Connection Properties window. In the Server name input box type your local db path - mine was “(LocalDB)\MSSQLLocalDB”. Next select a database in the Connect to a database section. Click the Test Connection button at the bottom left and you should be prompted with a success popup.

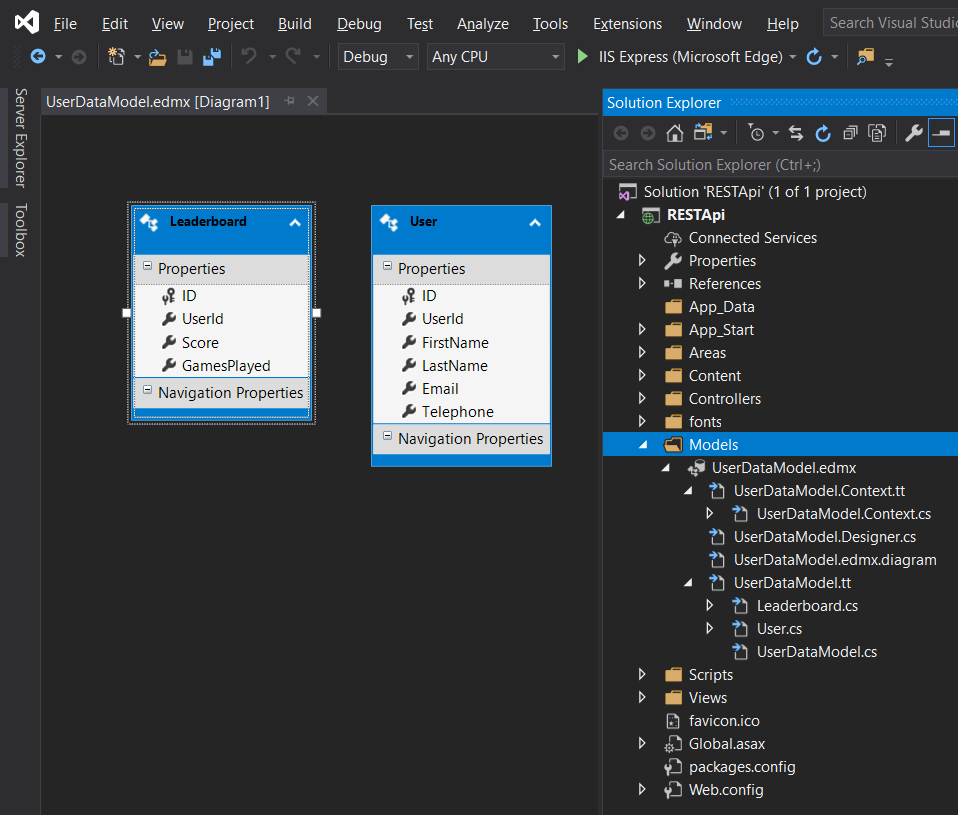


To find your local db information open command prompt (hold down the Windows Key and Press R on your keyboard then type cmd) and enter sqllocaldb info.

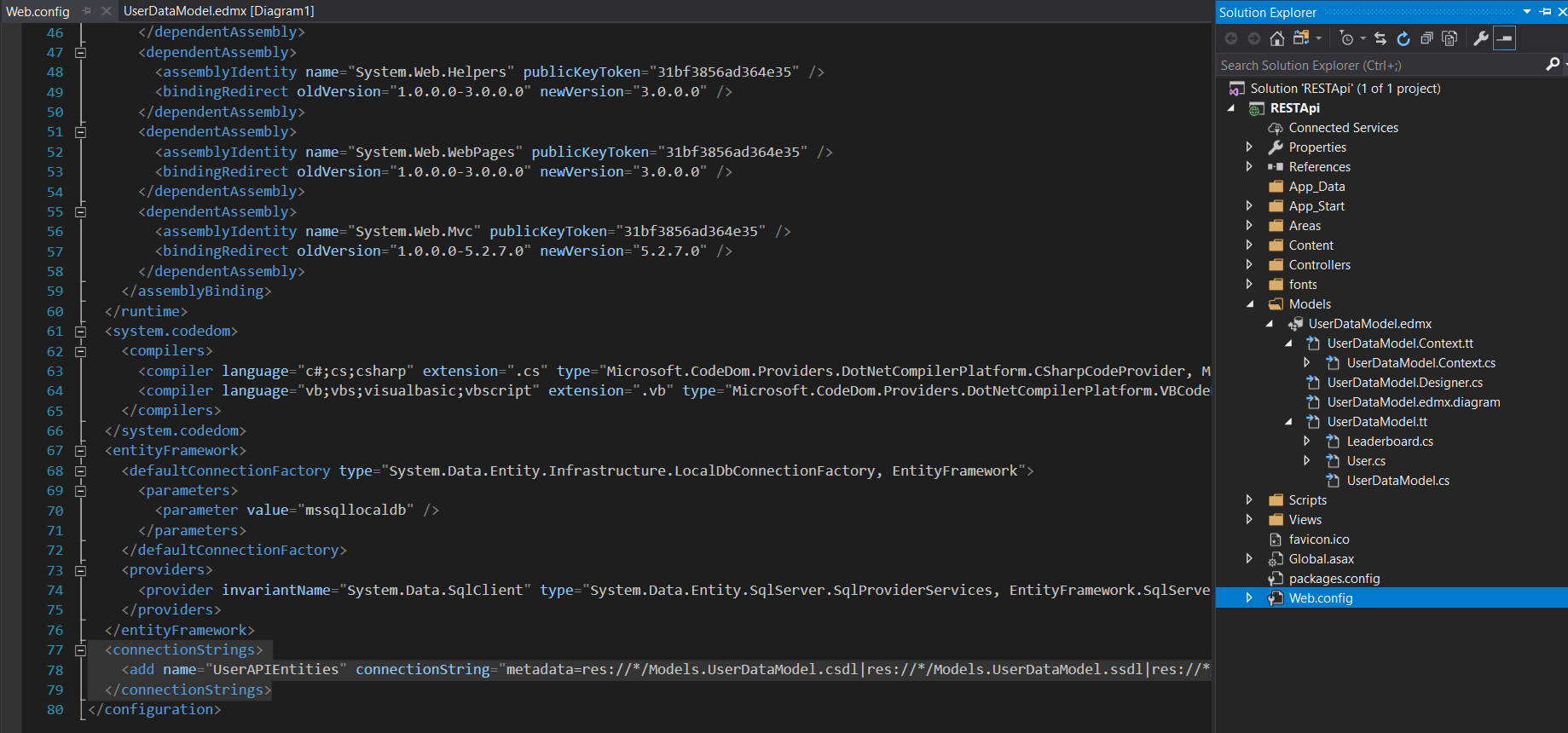
Once finished with the Connection Properties I was presented with a Connection string, the checkbox to “Save connection settings in Web.Config as:” checked and the name as “masterEntities”. Now click next.



Select the db objects you desire and click finish.



Once the Visual Studios has generated the entity data model you should have a .edmx file and a few other sub files.



In the Web.config file you should now have this connection in the connectionStrings tag.

One thing to note is that if you build the models via this method you should update the Users.cs and Leaderboard.cs model files to have display names clearer presentation in the UI. The files should look like the below where each getter and setter will have a name assigned, e.g. [Display(Name = "User Id")], to do so you would also need to include using System.ComponentModel.DataAnnotations; in the namespace.

