(T18)討論 CommandLine 執行 SqlMetal 自動生成 LinqToSql
CourseGUID: 5ba9a6fe-7475-4b0c-8b99-bbcf7f5e2e1c

==

(T18)討論 CommandLine 執行 SqlMetal 自動生成 LinqToSql

==

==

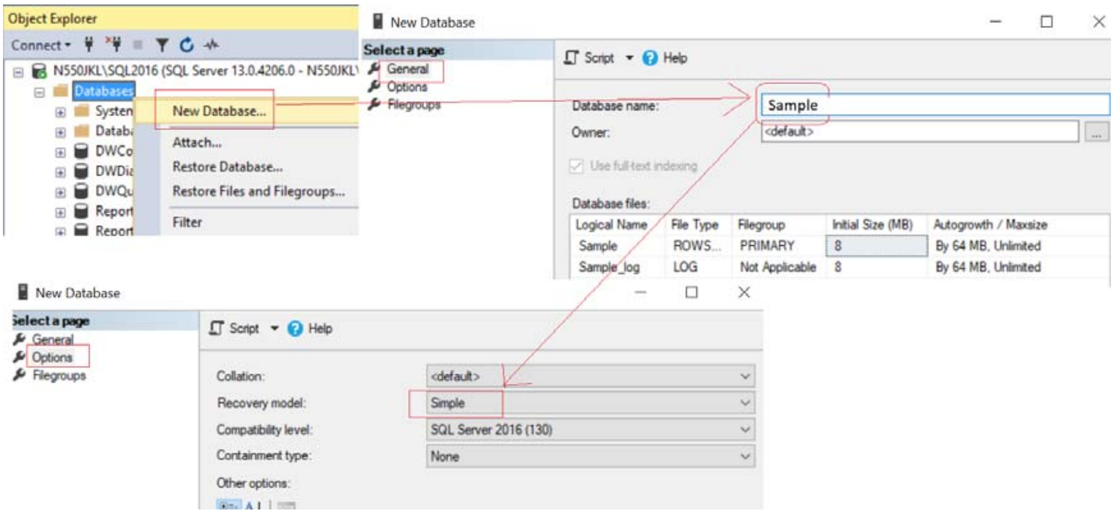
0. Introduction

In the previous tutorial, we use visual studio to create Linq to Sql classes.
This tutorial will demonstrate using SqlMetal to create Linq to Sql classes.

1. Web Form Application - Linq Query

1.1. TSQL

Database --> Right Click --> New Database -->
Database Name : Sample
Options --> Recovery Model : Simple



Database name: Sample
Owner: <default>
☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Sample	ROWS...	PRIMARY	8	By 64 MB, Unlimited
Sample_log	LOG	Not Applicable	8	By 64 MB, Unlimited

Collation: <default>
Recovery model: Simple
Compatibility level: SQL Server 2016 (130)
Containment type: None
Other options:

--Create a Sample DataBase and Run the following TSQL

```
/*
1.
One Team can have many Gamers
One Gamer can have One Team.
This is One to Many Relationship.
2.
Team Id==4 has no Gamer.
Gamer Id==7 has no Team.
*/
--1 ----------------------------------------------------------
--Drop Table if it exists.
--IF OBJECT_ID('Gamer') IS NOT NULL
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Gamer' ) )
    BEGIN
        TRUNCATE TABLE Gamer;
        DROP TABLE Gamer;
    END;
GO -- Run the previous command and begins new batch
--Drop Table if it exists.
--IF OBJECT_ID('Team') IS NOT NULL
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Team' ) )
    BEGIN
        TRUNCATE TABLE Team;
        DROP TABLE Team;
    END;
GO -- Run the previous command and begins new batch
--Create Tables
CREATE TABLE Team
    (
      Id INT PRIMARY KEY
             IDENTITY ,
      Name NVARCHAR(100) ,
      Type NVARCHAR(100)
    );
GO -- Run the previous command and begins new batch
CREATE TABLE Gamer
    (
      Id INT PRIMARY KEY
             IDENTITY ,
      Name NVARCHAR(50) ,
      Gender NVARCHAR(50) ,
      Score INT ,
      Type NVARCHAR(50) ,
      TeamId INT FOREIGN KEY REFERENCES Team ( Id )
    );
GO -- Run the previous command and begins new batch
--2 ----------------------------------------------------------
```

```sql
--Insert Data
INSERT  INTO Team
VALUES  ( 'Team1_Guardian', 'Guardian' );
INSERT  INTO Team
VALUES  ( 'Team2_Assassinator', 'Assassinator' );
INSERT  INTO Team
VALUES  ( 'Team3_Soldier', 'Soldier' );
INSERT  INTO Team
VALUES  ( 'Team4_Civilian', 'Civilian' );
GO -- Run the previous command and begins new batch
INSERT  INTO Gamer
VALUES  ( 'Name1 ABC', 'Male', 5000, 'Water', 1 );
INSERT  INTO Gamer
VALUES  ( 'Name2 ABCDE', 'Female', 4500, 'Fire', 3 );
INSERT  INTO Gamer
VALUES  ( 'Name3 EFGH', 'Male', 6500, 'Fire', 2 );
INSERT  INTO Gamer
VALUES  ( 'Name4 HIJKLMN', 'Female', 45000, 'Water', 2 );
INSERT  INTO Gamer
VALUES  ( 'Name5 NOP', 'Male', 3000, 'Wood', 1 );
INSERT  INTO Gamer
VALUES  ( 'Name6 PQRSTUVW', 'Male', 4000, 'Earth', 3 );
INSERT  INTO Gamer
VALUES  ( 'Name7 XYZ', 'Male', 4500, 'Metal', NULL );
GO -- Run the previous command and begins new batch
--3 ----------------------------------------------------------
--3.1. --------------------------------
--Drop Stored Procedure if it exists.
--IF OBJECT_ID('spGetGamers') IS NOT NULL
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spGetGamers' ) )
    BEGIN
        DROP PROCEDURE spGetGamers;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spGetGamers
AS
    BEGIN
        SELECT  Id ,
                Name ,
                Gender ,
                Score ,
                Type ,
                TeamId
        FROM    Gamer;
    END;
GO -- Run the previous command and begins new batch
--3.2. --------------------------------
-- Update Stored Procedure
--Drop Stored Procedure if it exists.
--IF OBJECT_ID('spInsertGamer') IS NOT NULL
IF ( EXISTS ( SELECT    *
             FROM      INFORMATION_SCHEMA.ROUTINES
             WHERE      ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spInsertGamer' ) )
    BEGIN
        DROP PROCEDURE spInsertGamer;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spInsertGamer
    @name NVARCHAR(50) ,
    @gender NVARCHAR(50) ,
    @score INT ,
    @type NVARCHAR ,
    @teamId int
AS
    BEGIN
        INSERT  INTO Gamer
        VALUES  ( @name, @gender, @score, @type, @teamId );
    END;
GO
--3.3. --------------------------------
-- Update Stored Procedure
--Drop Stored Procedure if it exists.
--IF OBJECT_ID('spUpdateGamer') IS NOT NULL
IF ( EXISTS ( SELECT    *
             FROM      INFORMATION_SCHEMA.ROUTINES
             WHERE      ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spUpdateGamer' ) )
    BEGIN
        DROP PROCEDURE spUpdateGamer;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spUpdateGamer
    @id INT ,
    @name NVARCHAR(50) ,
    @gender NVARCHAR(50) ,
    @score INT ,
    @type NVARCHAR ,
    @teamId int
AS
    BEGIN
        UPDATE  Gamer
        SET     Name = @name ,
                Gender = @gender ,
                Score = @score ,
                Type = @type ,
                TeamId = @teamId
        WHERE   Id = @id;
    END;
GO -- Run the previous command and begins new batch
--3.4. --------------------------------
-- Delete Stored Procedure
--Drop Stored Procedure if it exists.
--IF OBJECT_ID('spDeleteGamer') IS NOT NULL
```

```
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spDeleteGamer' ) )
    BEGIN
        DROP PROCEDURE spDeleteGamer;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spDeleteGamer @Id int
AS
    BEGIN
        DELETE  FROM dbo.Gamer
        WHERE   Id = @Id;
    END;
GO
--3.5. --------------------------------
--3.5.1. --------------------------------
-- Delete Stored Procedure
--Drop Stored Procedure if it exists.
--IF OBJECT_ID('spGetGamerByTeam') IS NOT NULL
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.ROUTINES
              WHERE     ROUTINE_TYPE = 'PROCEDURE'
                        AND LEFT(ROUTINE_NAME, 3) NOT IN ( 'sp_', 'xp_', 'ms_' )
                        AND SPECIFIC_NAME = 'spGetGamerByTeam' ) )
    BEGIN
        DROP PROCEDURE spGetGamerByTeam;
    END;
GO -- Run the previous command and begins new batch
CREATE PROCEDURE spGetGamerByTeam
    @teamId INT ,
    @teamName NVARCHAR(100) OUT
AS
    BEGIN
        SELECT  @teamName = Name
        FROM    dbo.Team
        WHERE   Id = @teamId;
        SELECT  *
        FROM    dbo.Gamer
        WHERE   TeamId = @teamId;
    END;
GO
--3.5.2. --------------------------------
--Test
DECLARE @teamName NVARCHAR(50);
EXECUTE spGetGamerByTeam 1, @teamName OUT;
SELECT  @teamName;
```

1.2. Set up SQL Authentication

In SQL server

Object Explorer --> Security --> Logins --> New Logins
-->
General Tab

Login Name :

Tester

Password:

1234

Default Database:

Sample

-->

Server Roles Tab

Select

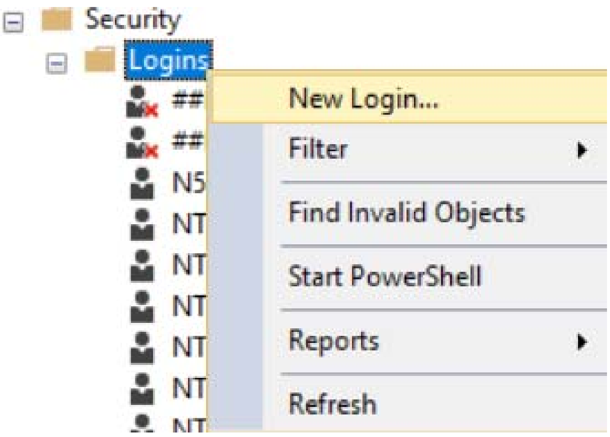
sysadmin

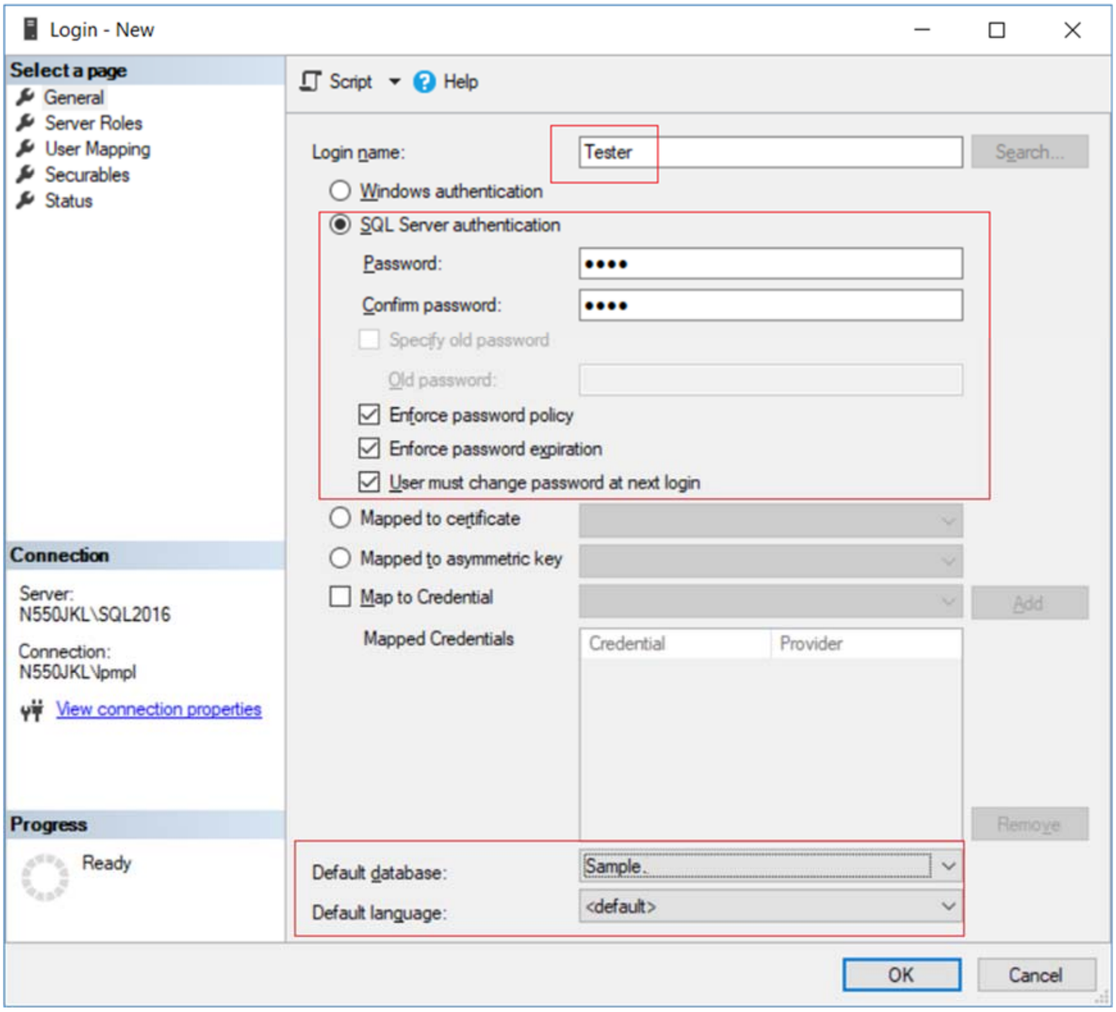
-->

User Mapping Tab

Select **Sample**

Select every Roles.





Login name: Tester

○ Windows authentication

● SQL Server authentication

Password: ••••

Confirm password: ••••

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

☑ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential

Default database: Sample

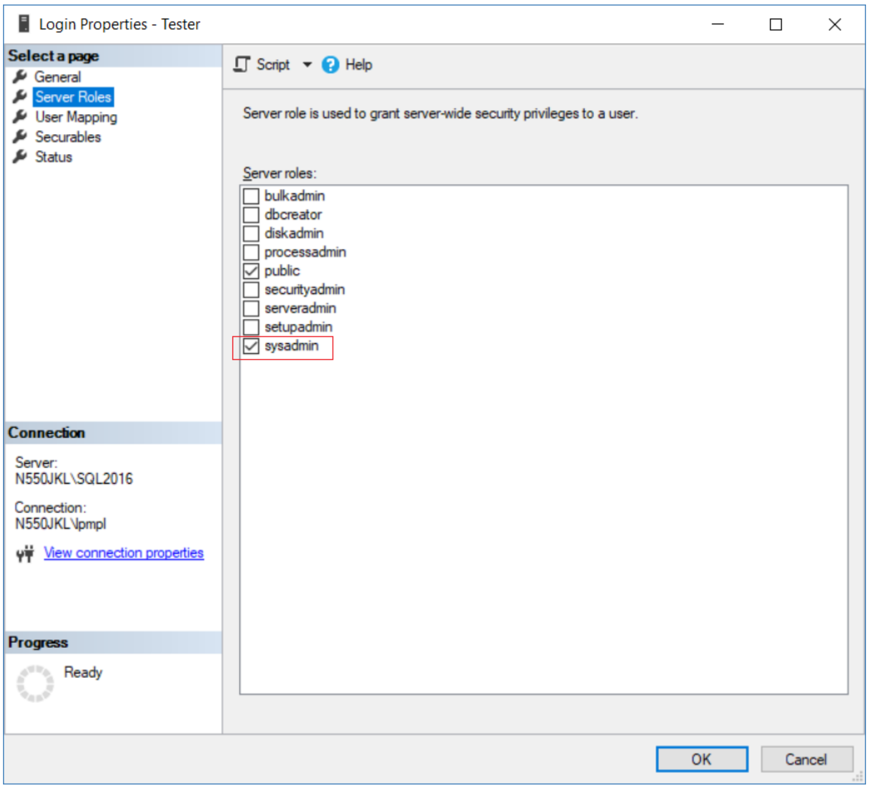
Default language: <default>

Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl

Ready



Server role is used to grant server-wide security privileges to a user.

Server roles:

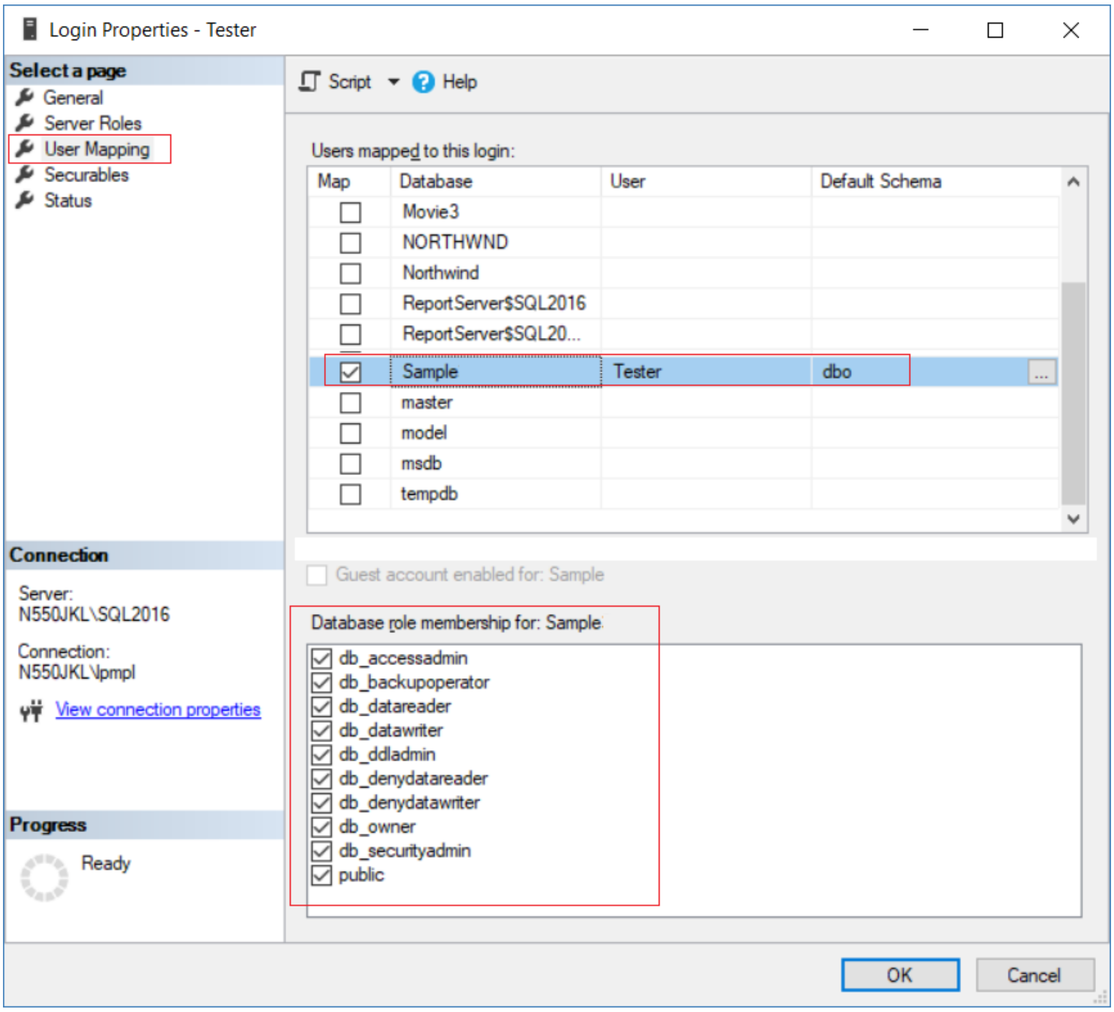
- □ bulkadmin
- □ dbcreator
- □ diskadmin
- □ processadmin
- ☑ public
- □ securityadmin
- □ serveradmin
- □ setupadmin
- ☑ sysadmin

Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\lpmpl

Ready



Users mapped to this login:

Map	Database	User	Default Schema
□	Movie3		
□	NORTHWND		
□	Northwind		
□	ReportServer$SQL2016		
□	ReportServer$SQL20...		
☑	Sample	Tester	dbo
□	master		
□	model		
□	msdb		
□	tempdb		

□ Guest account enabled for: Sample

Database role membership for: Sample

- ☑ db_accessadmin
- ☑ db_backupoperator
- ☑ db_datareader
- ☑ db_datawriter
- ☑ db_ddladmin
- ☑ db_denydatareader
- ☑ db_denydatawriter
- ☑ db_owner
- ☑ db_securityadmin
- ☑ public

Connection

Server:
N550JKL\SQL2016

Connection:
N550JKL\pmpl

View connection properties

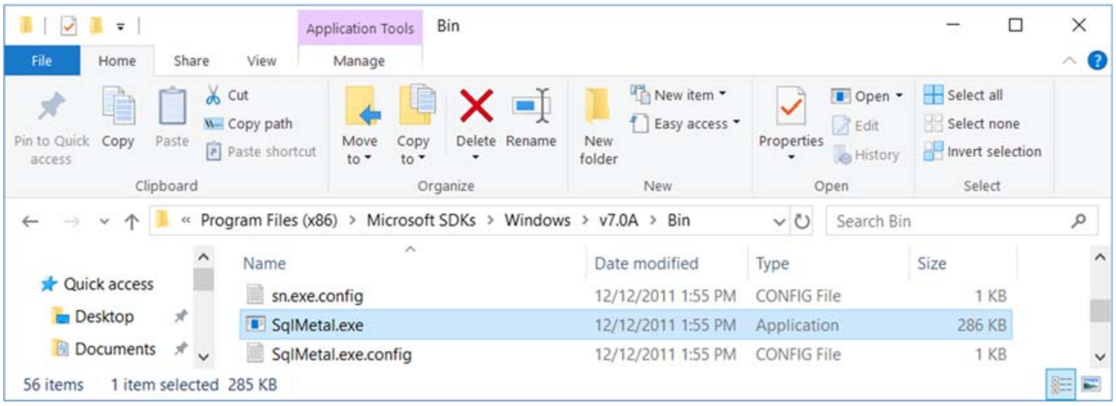
Progress

Ready

2. Using SqlMetal to create DBML

In the previous tutorial, we use visual studio to create Linq to Sql classes.
This tutorial will demonstrate using SqlMetal to create Linq to Sql classes.

SqlMetal is location in
C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\SqlMetal.exe
please do not double-click it.



« Program Files (x86) › Microsoft SDKs › Windows › v7.0A › Bin

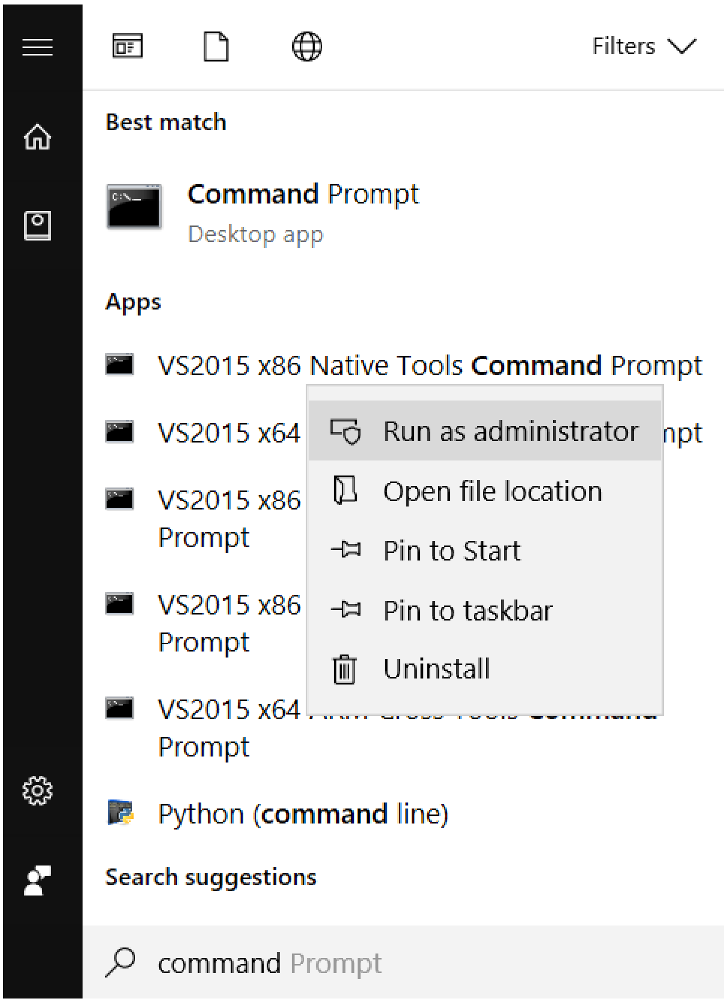
Name	Date modified	Type	Size
sn.exe.config	12/12/2011 1:55 PM	CONFIG File	1 KB
SqlMetal.exe	12/12/2011 1:55 PM	Application	286 KB
SqlMetal.exe.config	12/12/2011 1:55 PM	CONFIG File	1 KB

56 items   1 item selected 285 KB

Step1:
Create folder "**C:\LinqToSqlClasses**"

Step2:

Run "**Visual Studio Command prompt**" as an "**Administrator**"
In windows 10 -->
Start --> search "**command**" --> VS2015 x86 Native Tools Command Prompt
--> Right click --> Run as administrator



sqlmetal.exe /?
this command will show the options documentation.
If you want to know more, you can visit MSDN
http://msdn.microsoft.com/en-gb/library/vstudio/bb386987(v=vs.100).aspx

Step3:

SqlMetal.exe /server:N550JKL\SQL2016 /database:Sample /namespace:Sample /dbml:C:\LinqToSqlClasses\Sample.dbml /Context:SampleDataContext

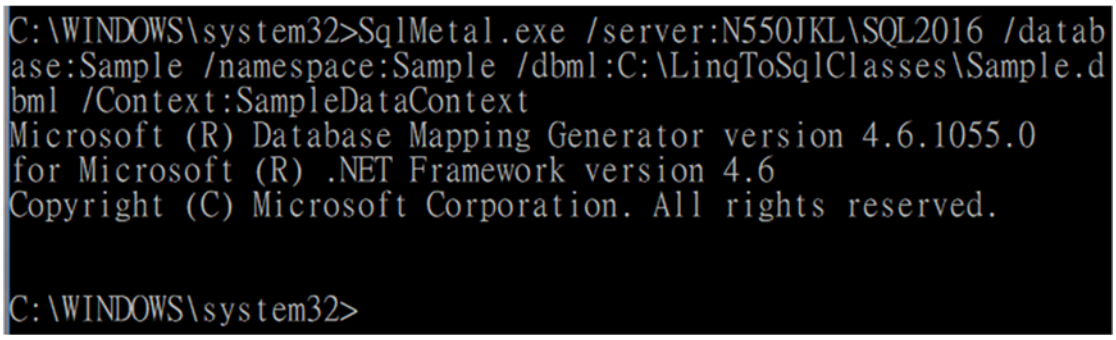
/server:N550JKL\SQL2016
This is the database server name on my current machine.

/database:Sample
I want to connect to "Sample" database.

/namespace:Sample
When we create Linq to Sql classes, the namespace is "**Sample**"

/dbml:C:\LinqToSqlClasses\Sample.dbml
I want to create **Sample.dbml** in **C:\LinqToSqlClasses**

/Context:SampleDataContext
When we create Linq to Sql classes, the data context class name is "**SampleDataContext**"

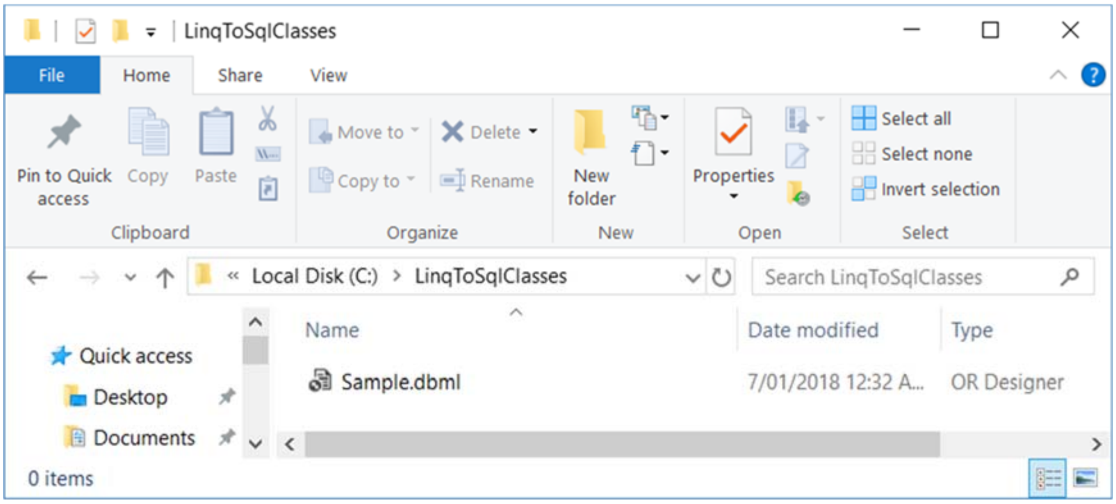


```
C:\WINDOWS\system32>SqlMetal.exe /server:N550JKL\SQL2016 /datab
ase:Sample /namespace:Sample /dbml:C:\LinqToSqlClasses\Sample.d
bml /Context:SampleDataContext
Microsoft (R) Database Mapping Generator version 4.6.1055.0
for Microsoft (R) .NET Framework version 4.6
Copyright (C) Microsoft Corporation. All rights reserved.


C:\WINDOWS\system32>
```

Step4:

Go and check if the dbml has been created.
C:\LinqToSqlClasses\Sample.dbml



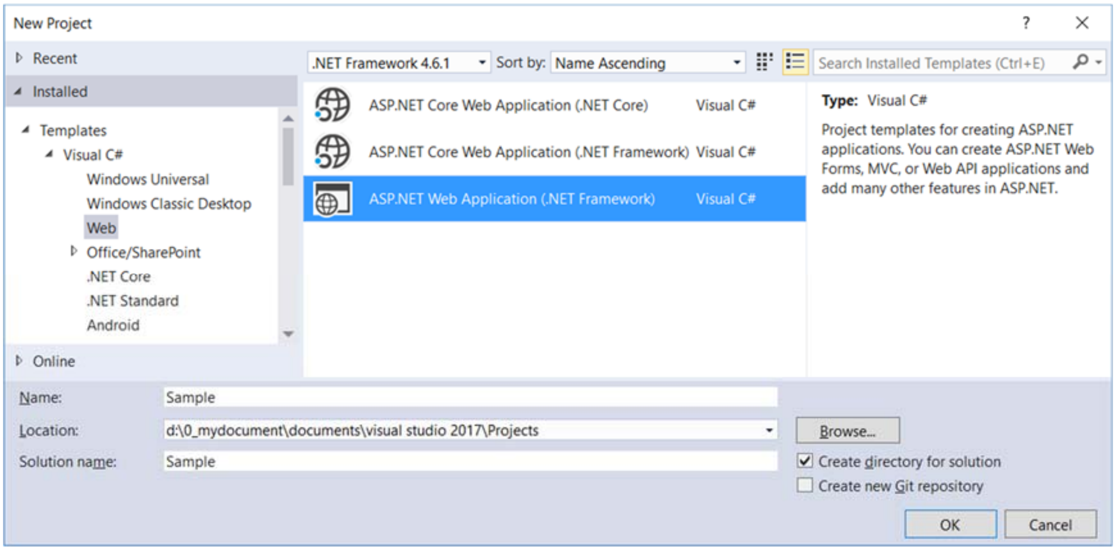
Name	Date modified	Type
Sample.dbml	7/01/2018 12:32 A...	OR Designer

3. Asp.Net Web Application with existing DBML

3.1. Create Web Application

Open Visual Studio, I am currently using VS2017
If you don't have it, you may follow the instruction here to download.
http://ithandyguytutorial.blogspot.com/2017/10/ch00install-visual-studio-2017-offline.html

New Project --> Web -->ASP.NET**Web Application (.Net Framework)**
-->
Name:
Sample
--> **Empty** --> Select "**Web Forms**" --> OK



.NET Framework 4.6.1 Sort by: Name Ascending

ASP.NET Core Web Application (.NET Core)	Visual C#
ASP.NET Core Web Application (.NET Framework)	Visual C#
ASP.NET Web Application (.NET Framework)	Visual C#

Type: Visual C#

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

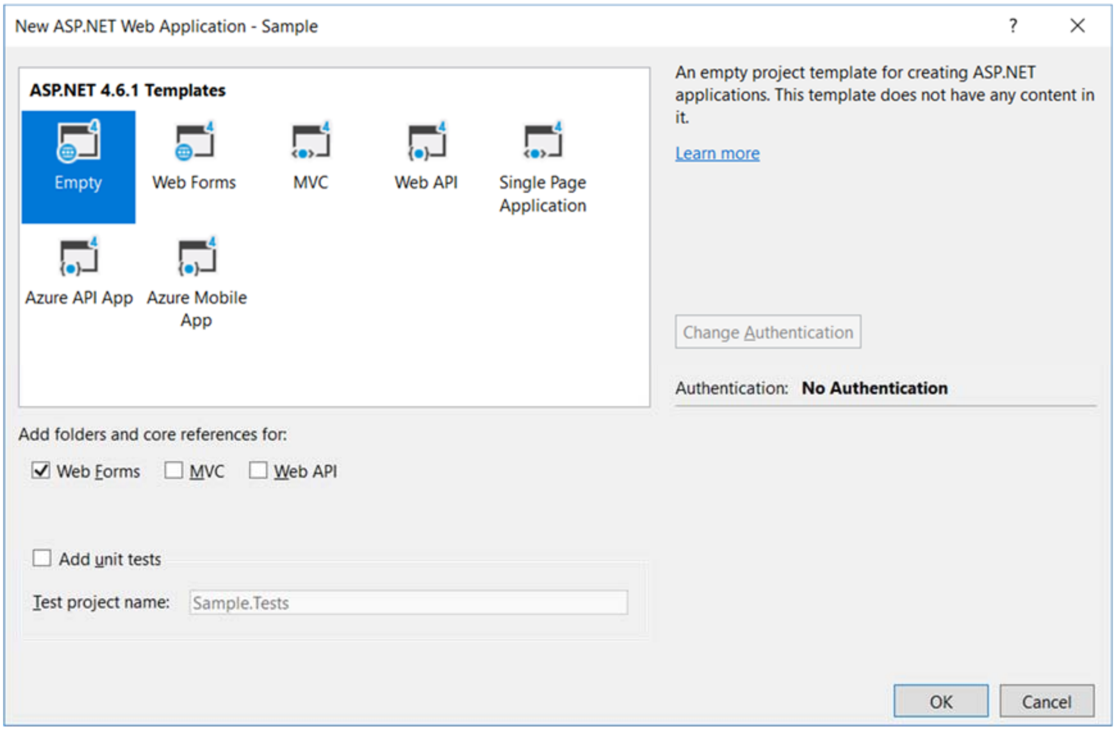
Name: Sample

Location: d:\0_mydocument\documents\visual studio 2017\Projects

Solution name: Sample

☑ Create directory for solution

☐ Create new Git repository



ASP.NET 4.6.1 Templates

Empty, Web Forms, MVC, Web API, Single Page Application, Azure API App, Azure Mobile App

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Learn more

Change Authentication

Authentication: **No Authentication**

Add folders and core references for:

☑ Web Forms ☐ MVC ☐ Web API

☐ Add unit tests

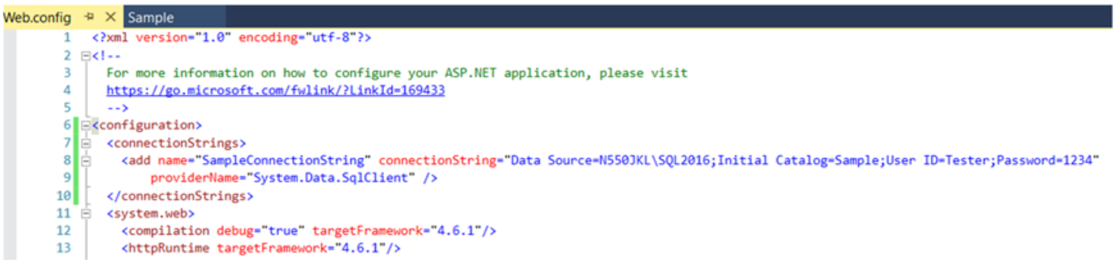
Test project name: Sample.Tests

3.2. Create Web Application

Add connection String

If you use Linq to Sql, you don't have to set this connection string.
I personally already get used to set it by myself.

```
<configuration>
  <connectionStrings>
    <add name="SampleConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial
Catalog=Sample;User ID=Tester;Password=1234"
       providerName="System.Data.SqlClient" />
  </connectionStrings>
```

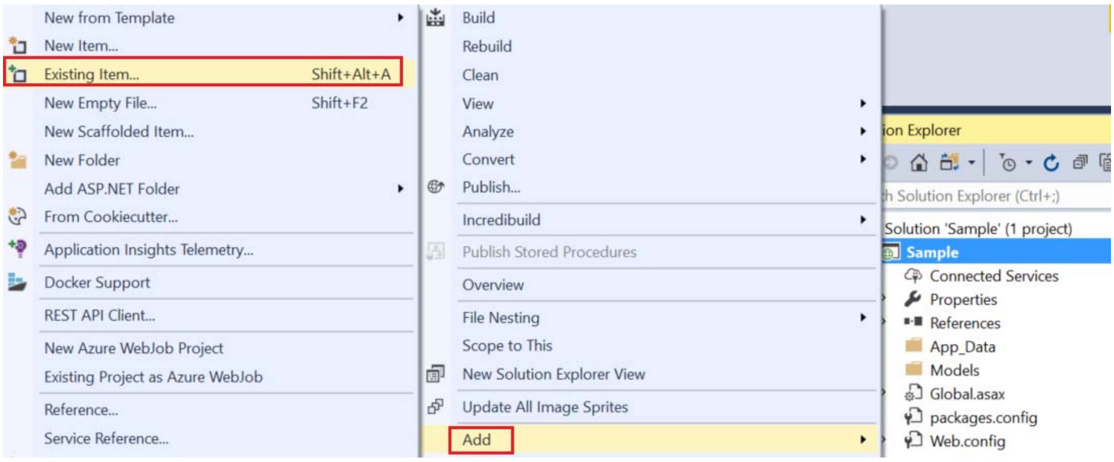


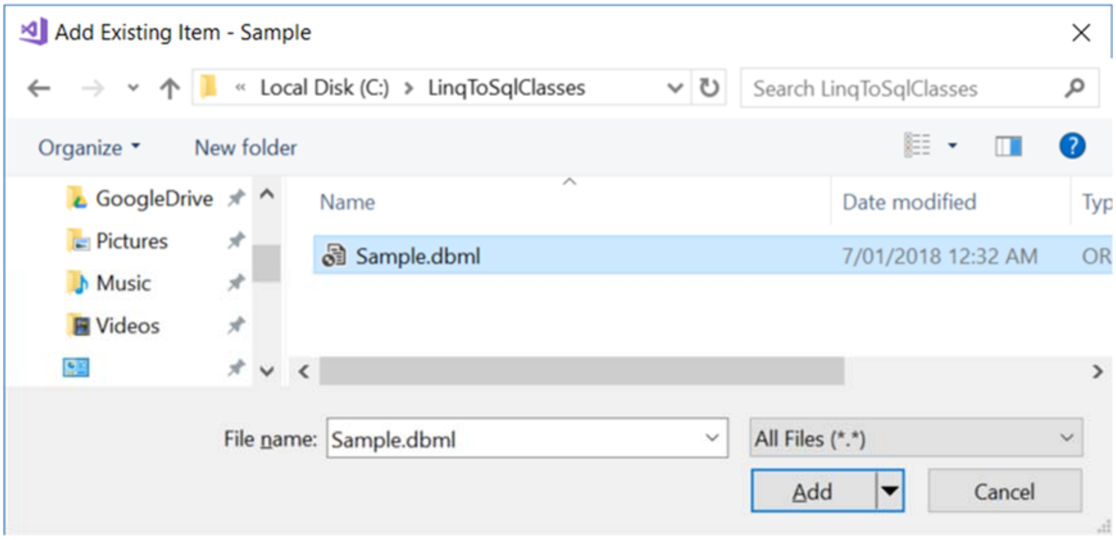
```xml
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
  <connectionStrings>
    <add name="SampleConnectionString" connectionString="Data Source=N550JKL\SQL2016;Initial Catalog=Sample;User ID=Tester;Password=1234"
        providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.6.1"/>
    <httpRuntime targetFramework="4.6.1"/>
```

3.3. Add existing DBML

Project Name --> Right click --> Add --> Existing Item...
--> select
C:\LinqToSqlClasses\Sample.dbml

When add existing dbml, it will copy that dbml to the project folder.
After you add dbml, it will generate the classes.





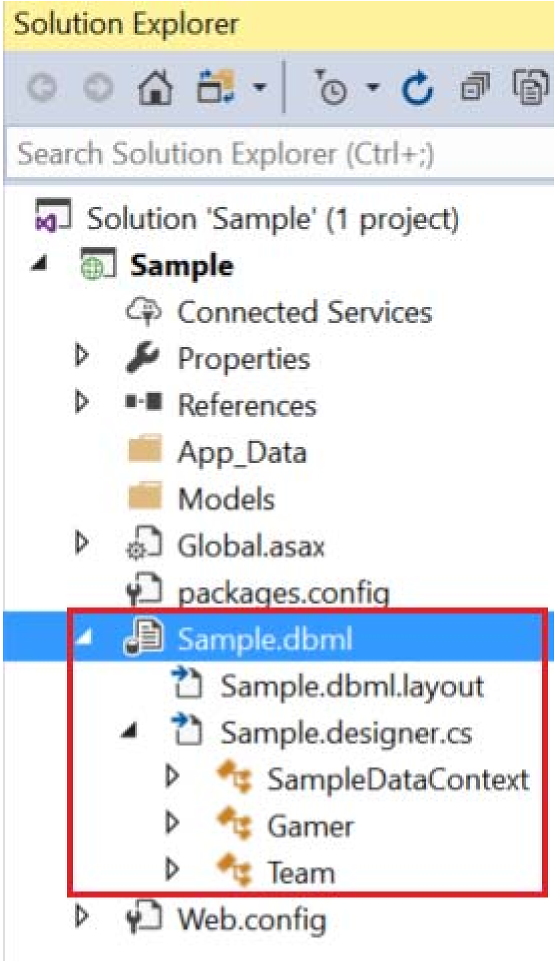
Add Existing Item - Sample

Local Disk (C:) > LinqToSqlClasses

Name	Date modified	Typ
Sample.dbml	7/01/2018 12:32 AM	OR

File name: Sample.dbml

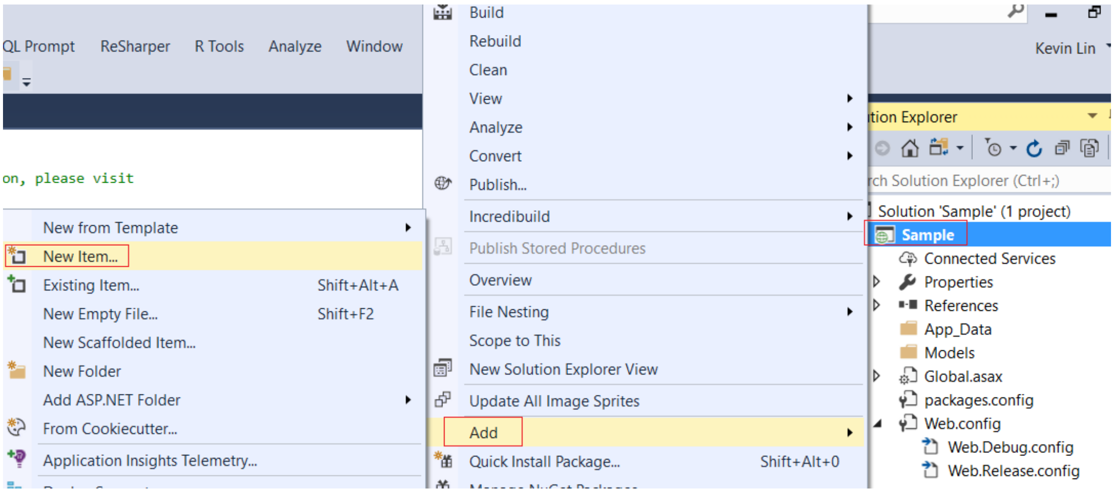
All Files (*.*)

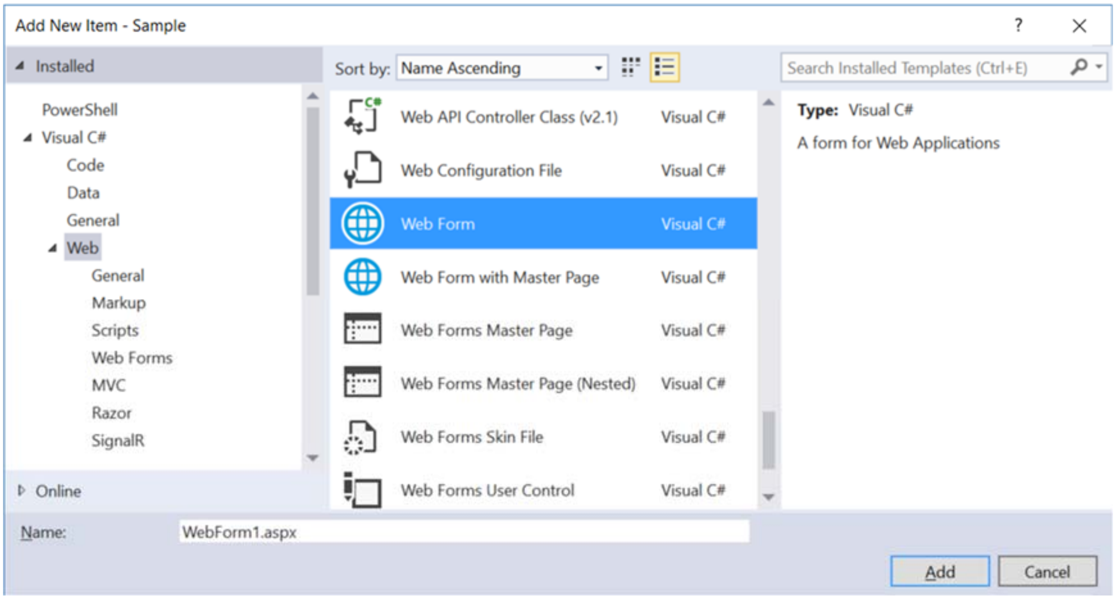


3.4. WebForm1.aspx

3.4.1. WebForm1.aspx

ProjectName --> Right Click --> Add --> New Item...
-->
WebForm
Name :
WebForm1.aspx





Add New Item - Sample
Installed
Sort by: Name Ascending
Search Installed Templates (Ctrl+E)
PowerShell
Visual C#
Code
Data
General
Web
General
Markup
Scripts
Web Forms
MVC
Razor
SignalR
Online
Web API Controller Class (v2.1) Visual C#
Web Configuration File Visual C#
Web Form Visual C#
Web Form with Master Page Visual C#
Web Forms Master Page Visual C#
Web Forms Master Page (Nested) Visual C#
Web Forms Skin File Visual C#
Web Forms User Control Visual C#
Type: Visual C#
A form for Web Applications
Add
Cancel

Name: WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Sample.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server"></asp:GridView>
        </div>
    </form>
</body>
</html>
```

3.4.2. WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Sample
{
    public partial class WebForm1 : System.Web.UI.Page
     {
        protected void Page_Load(object sender, EventArgs e)
         {
             GetData();
         }
        private void GetData()
         {
            string cs = ConfigurationManager.
               ConnectionStrings["SampleConnectionString"].ConnectionString;
            using (SampleDataContext dbContext = new SampleDataContext(cs))
            {
                IQueryable<Gamer> gamerQueryable =
                    from gamer in dbContext.Gamer
                    select gamer;
                GridView1.DataSource = gamerQueryable;
                GridView1.DataBind();
            }
        }
    }
}
```