(T3)討論 Api 的 HttpPost、HttpGet、HttpDelete 屬性。討論 Api 的 Put 的 HttpPut、FromBody、FromURI 屬性
CourseGUID 4c5822ff-7111-4e25-a336-ef18d48d54bd
==========
(T3)討論 Api 的 HttpPost、HttpGet、HttpDelete 屬性。討論 Api 的 Put 的 HttpPut、FromBody、FromURI 屬性
(T3-1)自動生成 Api 的 Post、Get、Put、Delete 對應到資料庫的 CRUD
(T3-2)討論 Api 的 Get 的 HttpGet 屬性
(T3-3)討論 Api 的 Post 的 HttpPost 屬性
(T3-4)討論 Api 的 Put 的 HttpPut、FromBody、FromURI 屬性
(T3-5)討論 Api 的 Delete 的 HttpDelete 屬性
==========

==========

1. OnlineGame DB

The tutorial will discuss
Auto-generate the API with Get、Post、Put、Delete
and then Read, Insert, Update, Delete data from the database
About HttpGet、HttpPost、HttpPut、HttpDelete.
About FromBody and FromURI

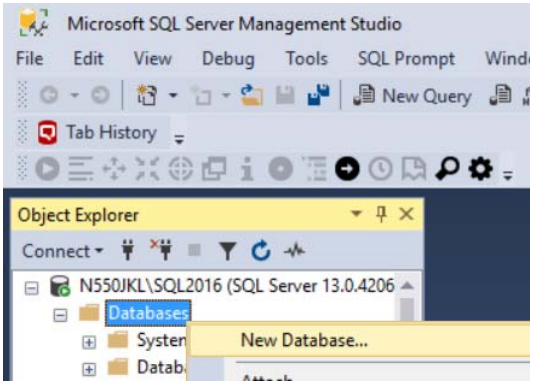
本堂課討論
建立一個 API with Get、Post、Put、Delete 並且 Read, Insert, Update, Delete data from the database。
關於 HttpGet、HttpPost、HttpPut、HttpDelete 四大屬性
關於 FromBody 和 FromURI

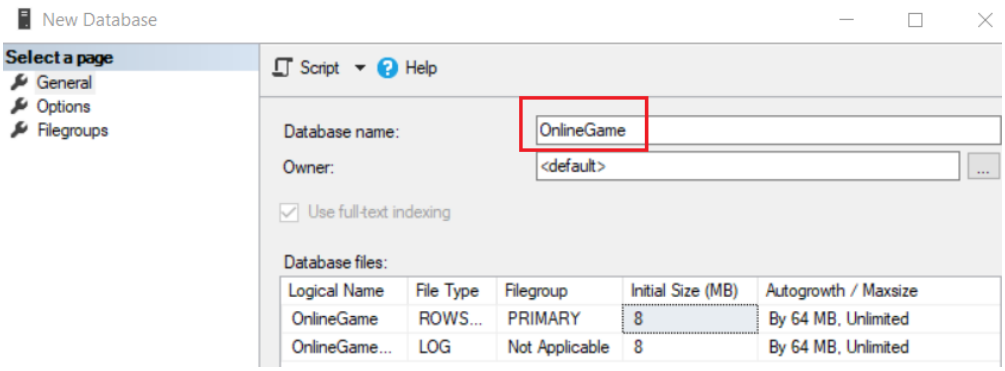
1.0. Some points

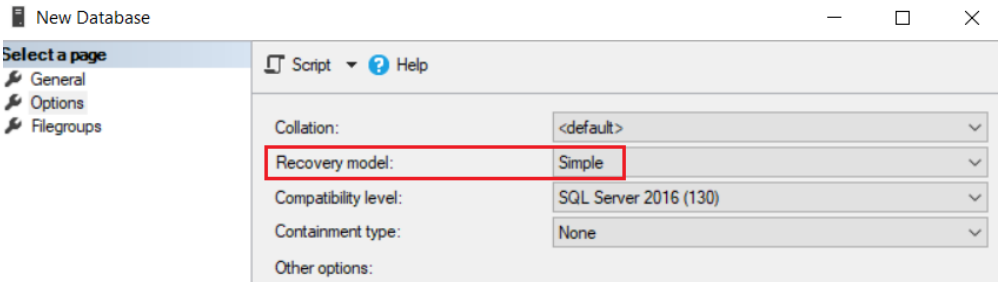
1.
Regular expression
https://regexr.com/
2.
Calling Stored Procedure from Entity Framework 6 Code First
http://www.dotnetodyssey.com/2015/03/12/calling-stored-procedure-from-entity-framework-6-code-first/

1.1. TSQL

In SQL server Management Studio (SSMS)
Database --> Right Click --> New Database -->
In General Tab -->
Name: **OnlineGame**
In options Tab --> Recovery model : **Simple**





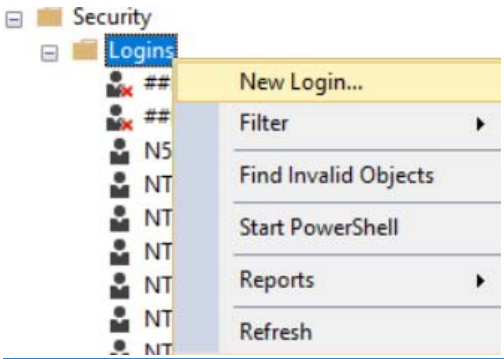


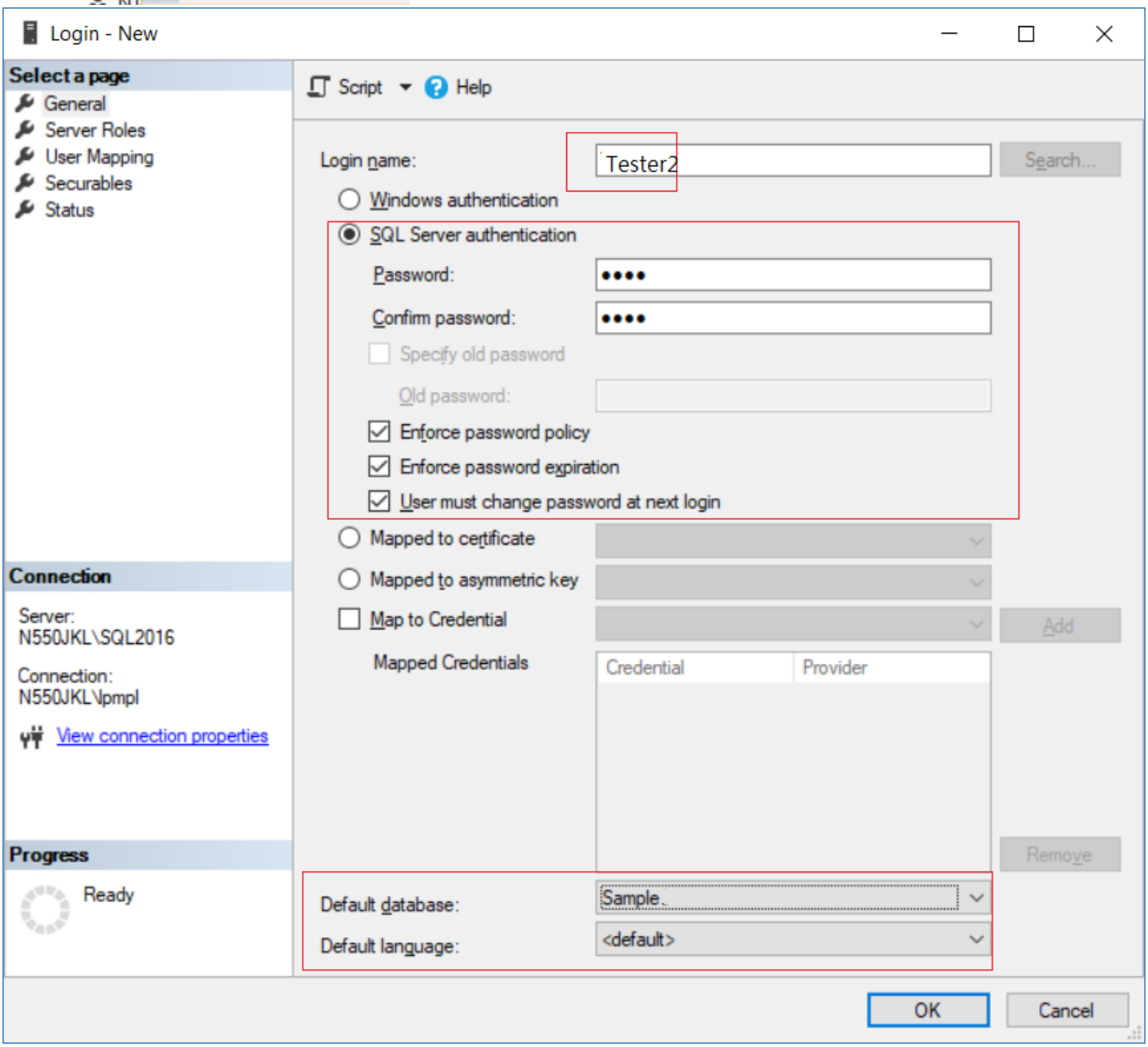

```
--1 ----------------------------------------------------------
--Drop Table if it exists.
IF ( EXISTS ( SELECT    *
              FROM      INFORMATION_SCHEMA.TABLES
              WHERE     TABLE_NAME = 'Gamer' ) )
   BEGIN
       TRUNCATE TABLE Gamer;
       DROP TABLE Gamer;
   END;
GO -- Run the previous command and begins new batch
--2 ----------------------------------------------------------
```

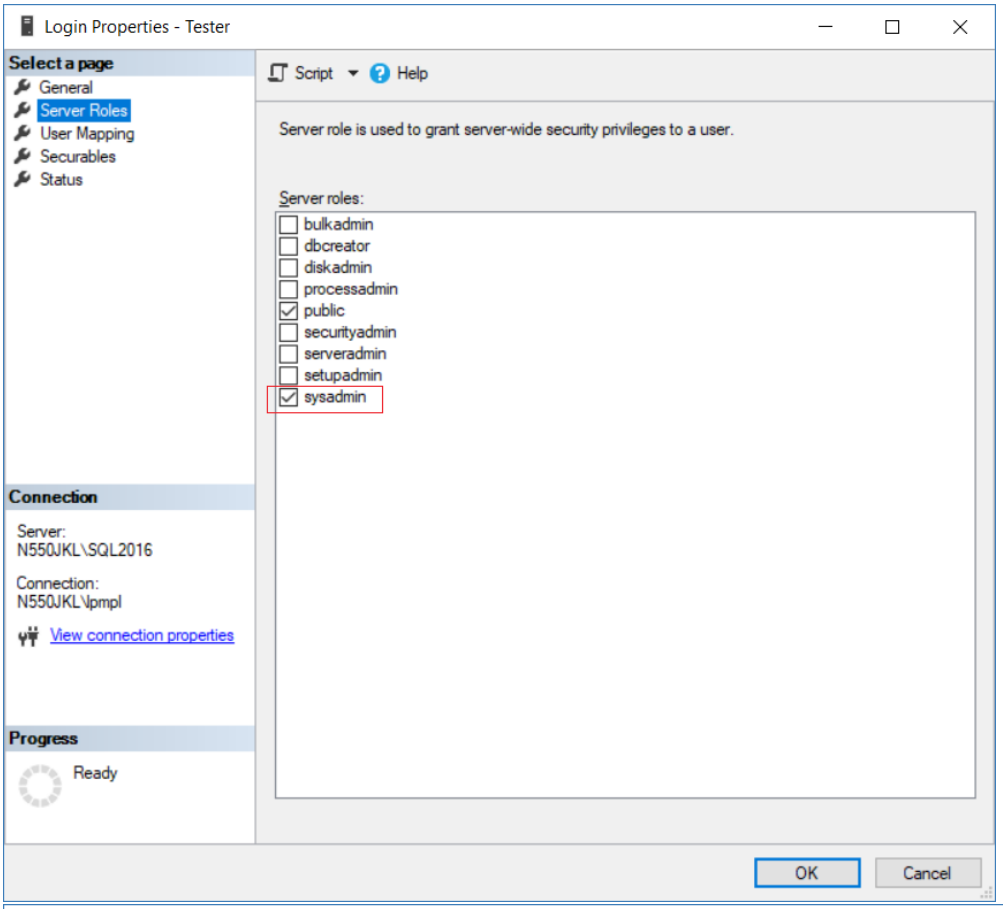
```
CREATE TABLE Gamer
    (
      Id INT PRIMARY KEY
             IDENTITY(1, 1)
             NOT NULL ,
      Name NVARCHAR(50) NOT NULL ,
      Gender NVARCHAR(50) NOT NULL ,
      Score INT NOT NULL ,
      GameMoney INT NOT NULL
    );
GO -- Run the previous command and begins new batch
--3 ----------------------------------------------------------
INSERT  INTO Gamer
VALUES  ( 'NameOne ABC', 'Male', 5000, 550 );
INSERT  INTO Gamer
VALUES  ( 'NameTwo ABCDE', 'Female', 4500, 1200 );
INSERT  INTO Gamer
VALUES  ( 'NameThree EFGH', 'Male', 6500, 3050 );
INSERT  INTO Gamer
VALUES  ( 'NameFour HIJKLMN', 'Female', 45000, 450 );
INSERT  INTO Gamer
VALUES  ( 'NameFive NOP', 'Male', 3000, 200 );
INSERT  INTO Gamer
VALUES  ( 'NameSix PQRSTUVW', 'Male', 4000, 700 );
INSERT  INTO Gamer
VALUES  ( 'NameSeven XYZ', 'Male', 450, 1500 );
GO -- Run the previous command and begins new batch
```

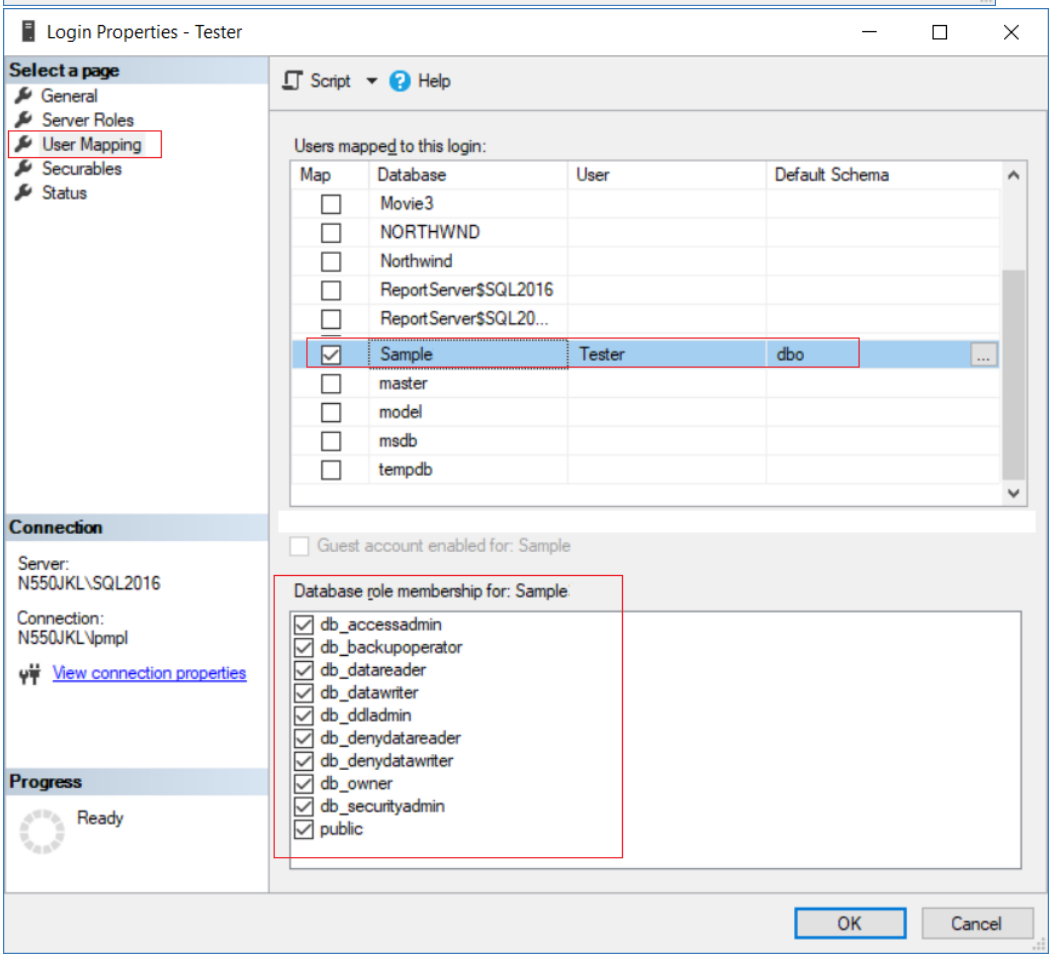
1.2. Security login

In SQL server
Object Explorer --> Security --> Logins --> New Logins
-->
General Tab
Login Name :
Tester2
Password:
1234
Default Database:
OnlineGame
-->
Server Roles Tab
Select
sysadmin
-->
User Mapping Tab
Select **OnlineGame**
Select every single role.


Security
Logins
New Login...
Filter
Find Invalid Objects
Start PowerShell
Reports
Refresh


Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Tester2
Search...
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
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
Sample
Default language:
<default>
Connection
Server:
N550JKL\SQL2016
Connection:
N550JKL\lpmpl
View connection properties
Progress
Ready
OK
Cancel





2. OnlineGame Solution

2.1. OnlineGame Solution

File --> New --> Project... -->

Other Project Types --> Visual Studio Solutions --> Blank Solution
-->

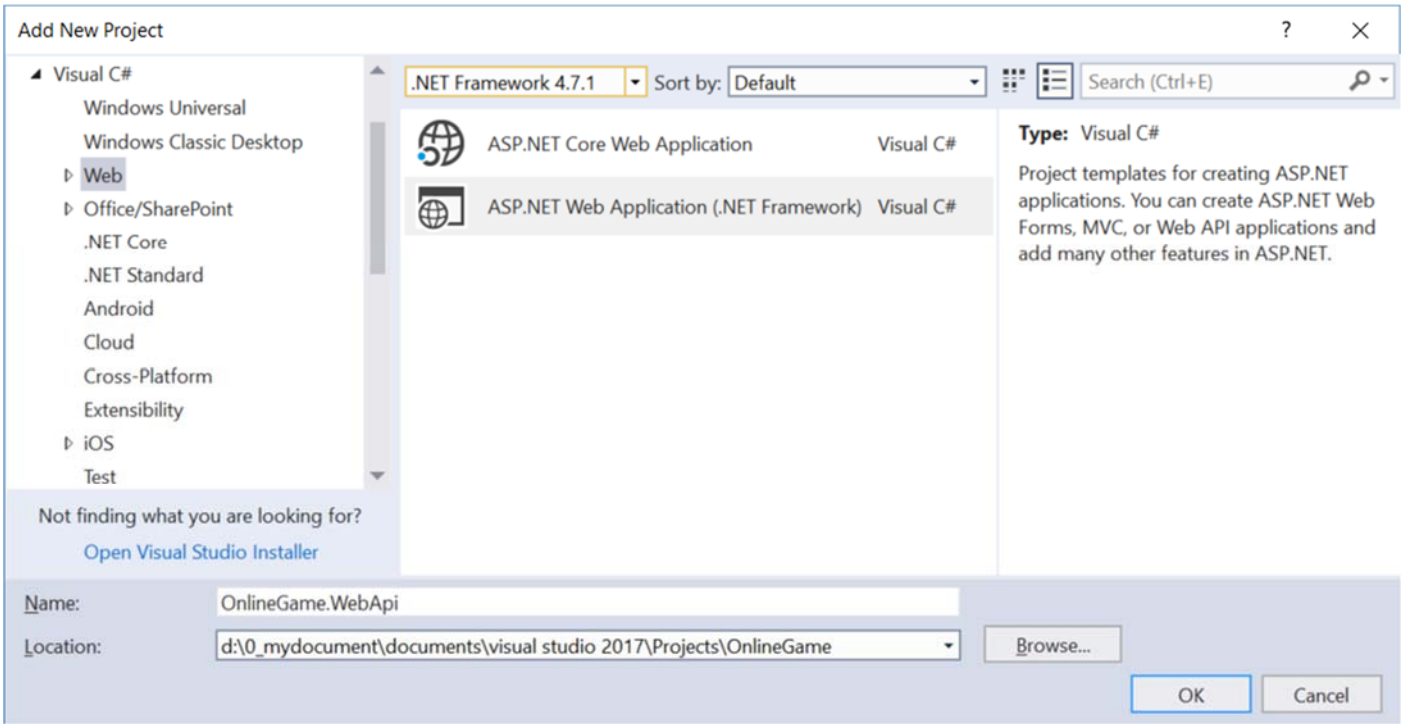
Name: **OnlineGame**

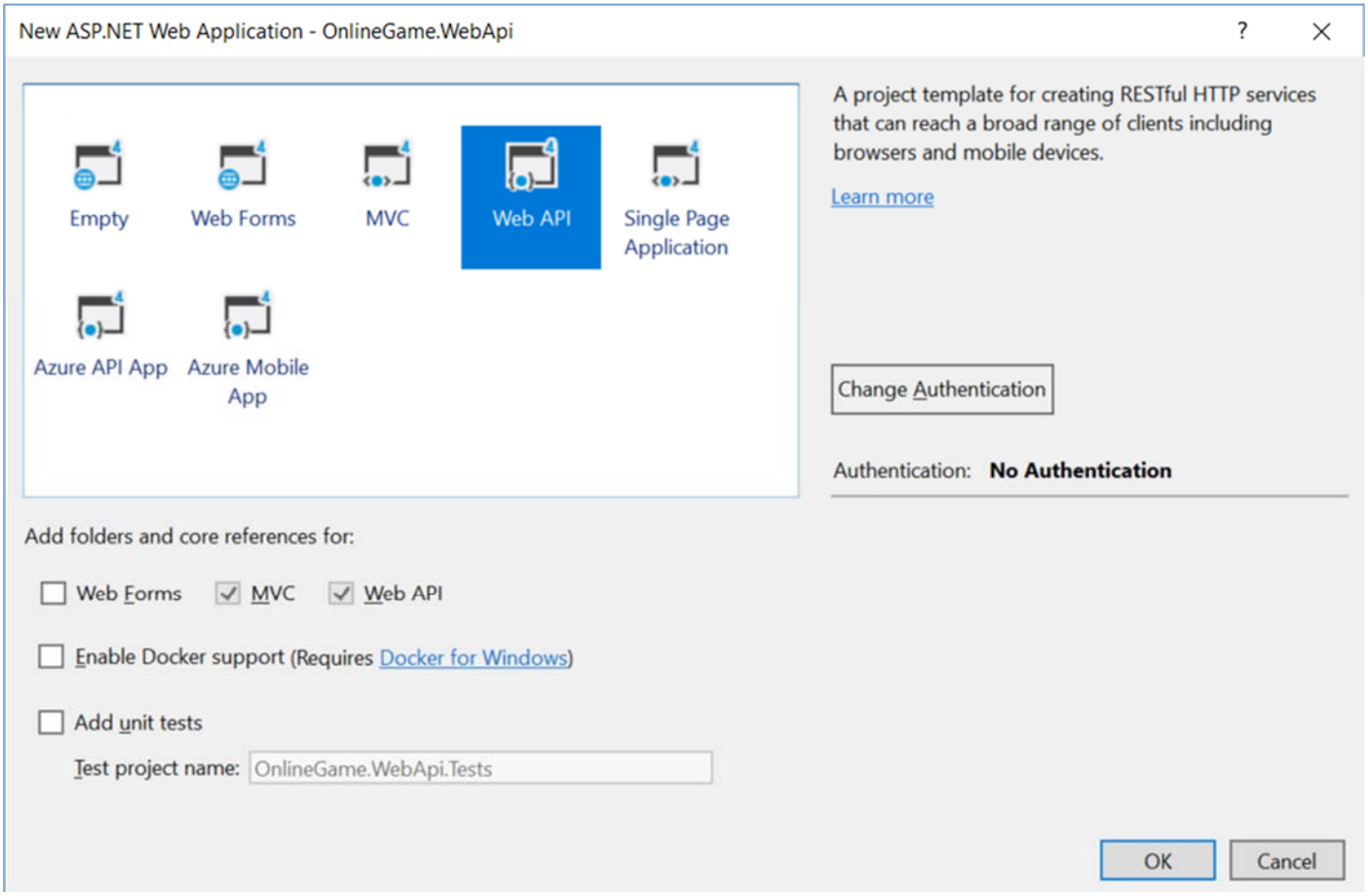
2.2. OnlineGame.WebApi

Solutions Name --> Add --> New Project -->

Visual C# --> Web --> ASP.NET Web Application (.Net Framework)
-->

Name: **OnlineGame.WebApi**
--> Select "**Web API**" --> OK





2.3. OnlineGame.Data

Solutions Name --> Add --> New Project -->

Visual C# --> **Class Library (.NET Framework)**

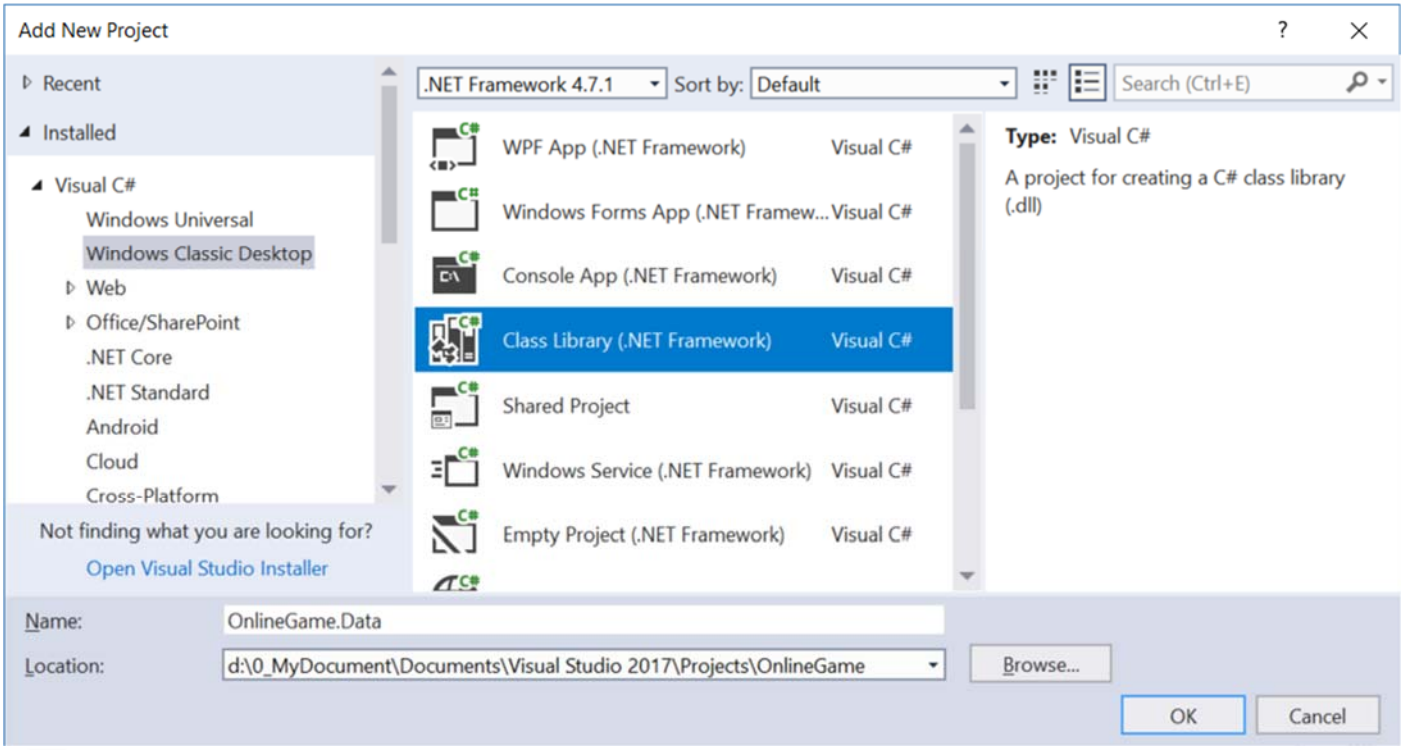
-->

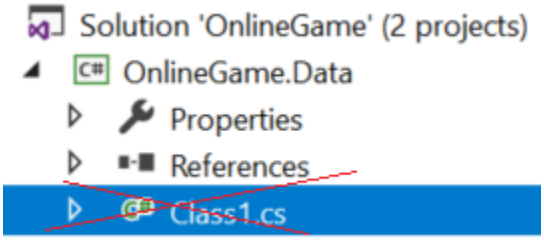
Name:

OnlineGame.Data

-->

Delete Class1.cs

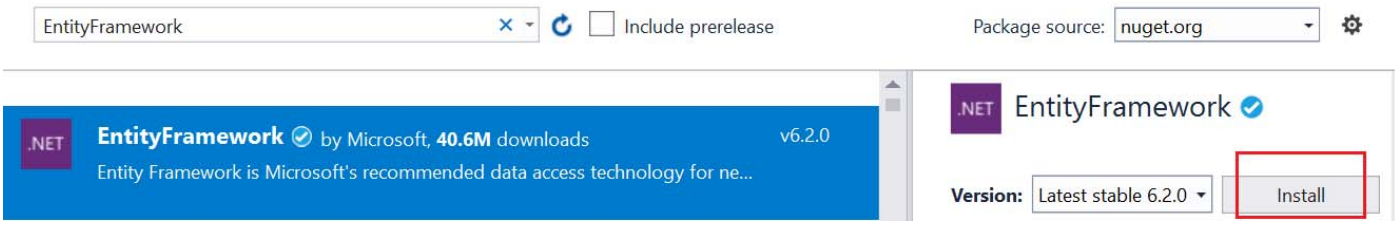




3. OnlineGame.Data

3.1. Install Entity Framework

Tools --> NuGet Package Manager --> Manage NuGet Packages for Solutions...
--> Browse tab --> Search : **EntityFramework**
--> Install it



3.2. ADO.Net Entity Data Model - Entity Framework

In Visual Studio 2017
Project Name --> Right Click --> Add --> New Item
--> Visual C# --> Data --> ADO.Net Entity Data Model
Name:
OnlineGameDataModel
-->

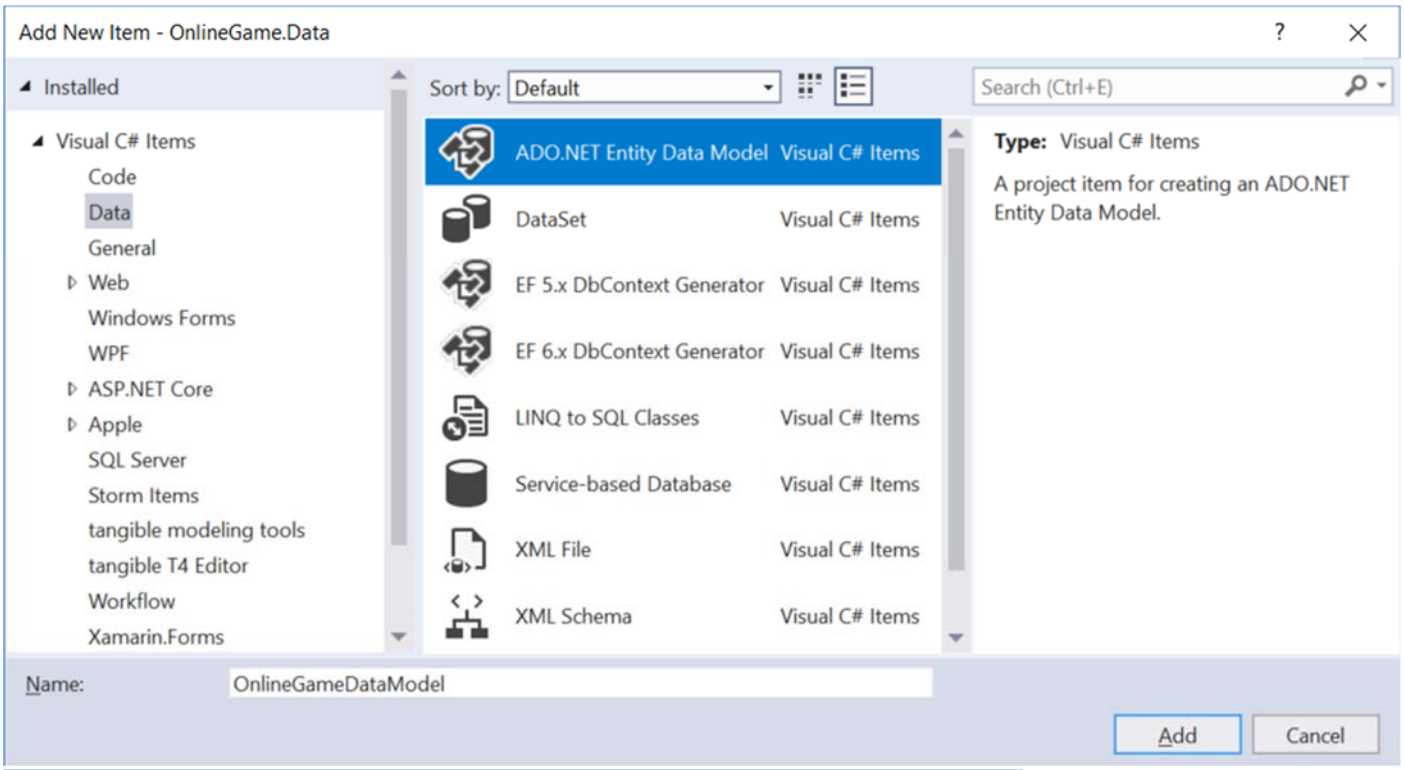
EF Designer from database

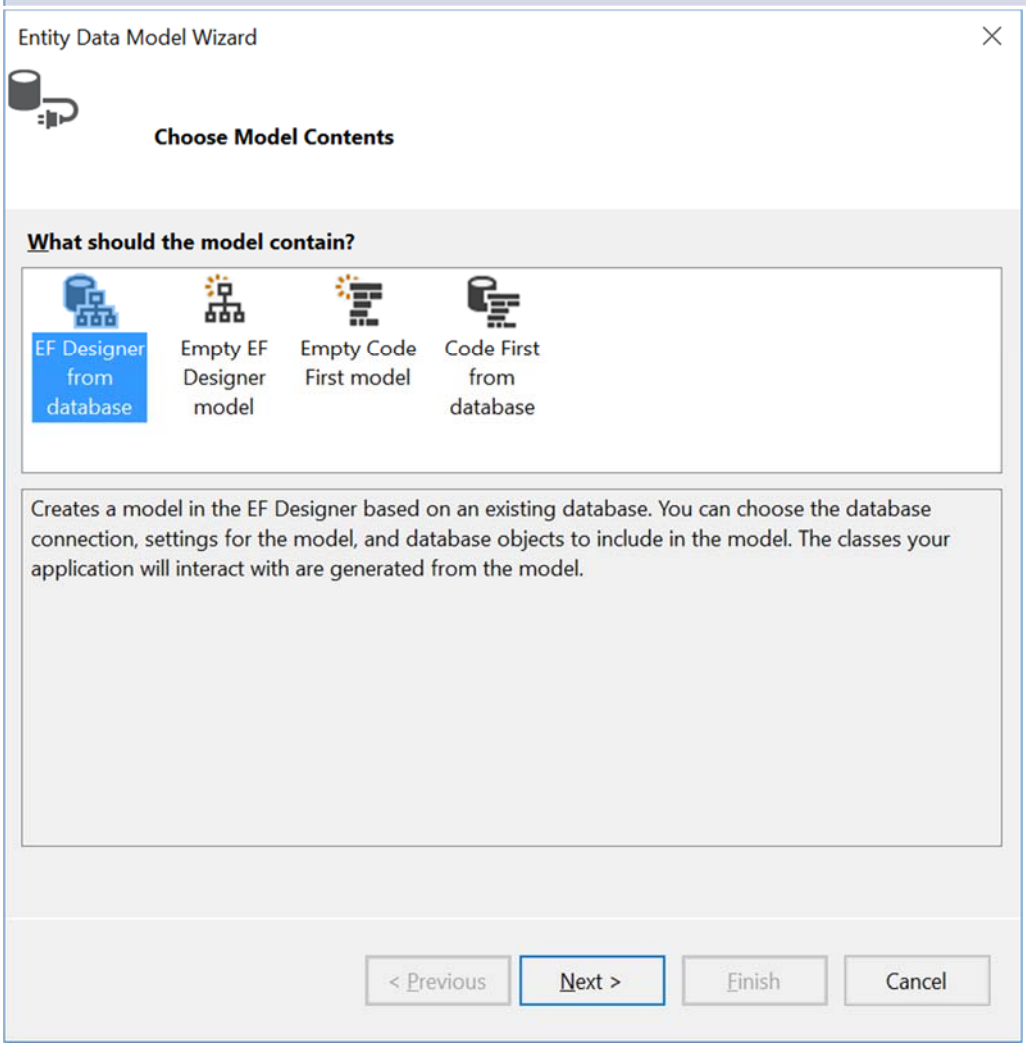
....

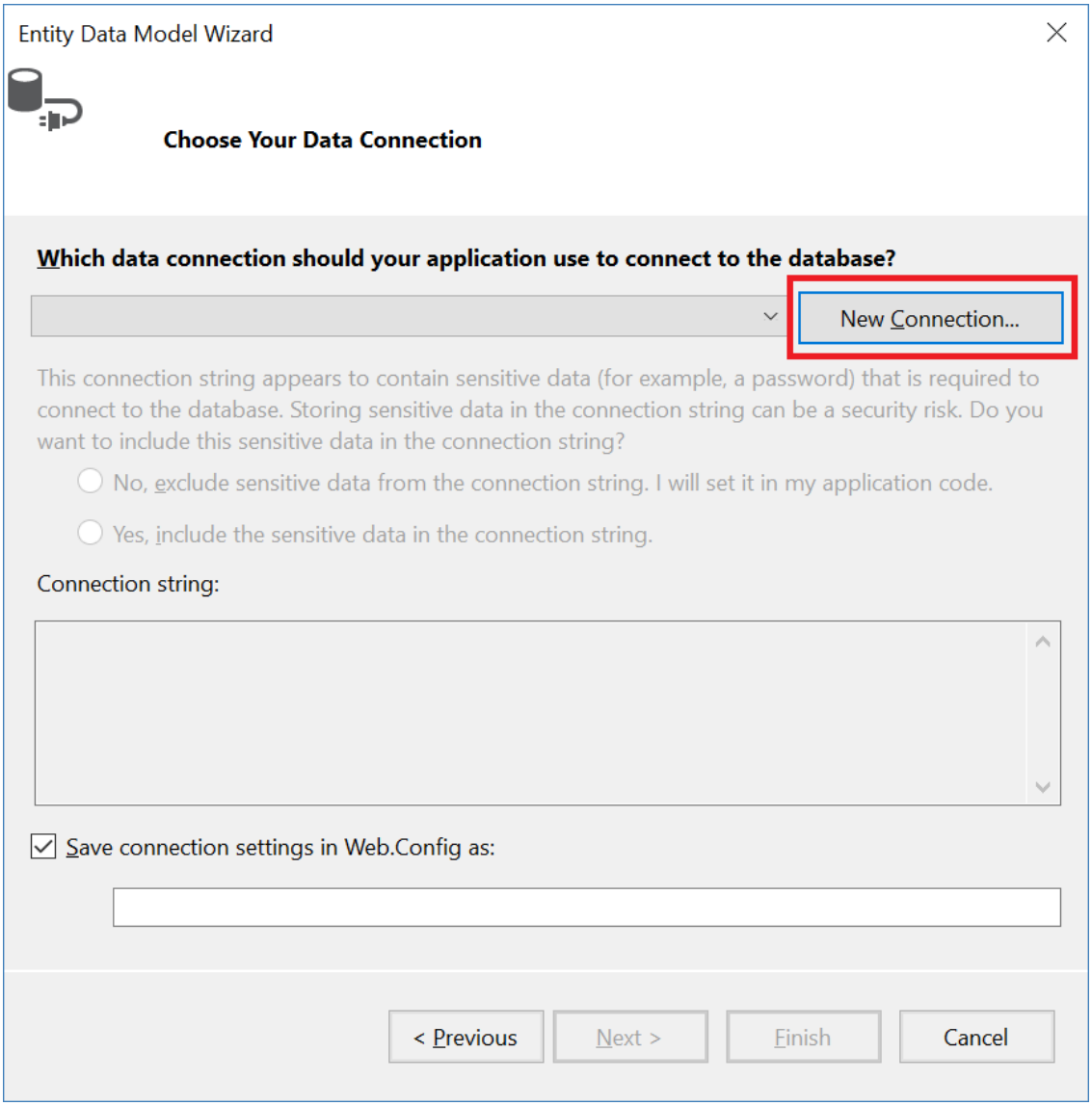
-->

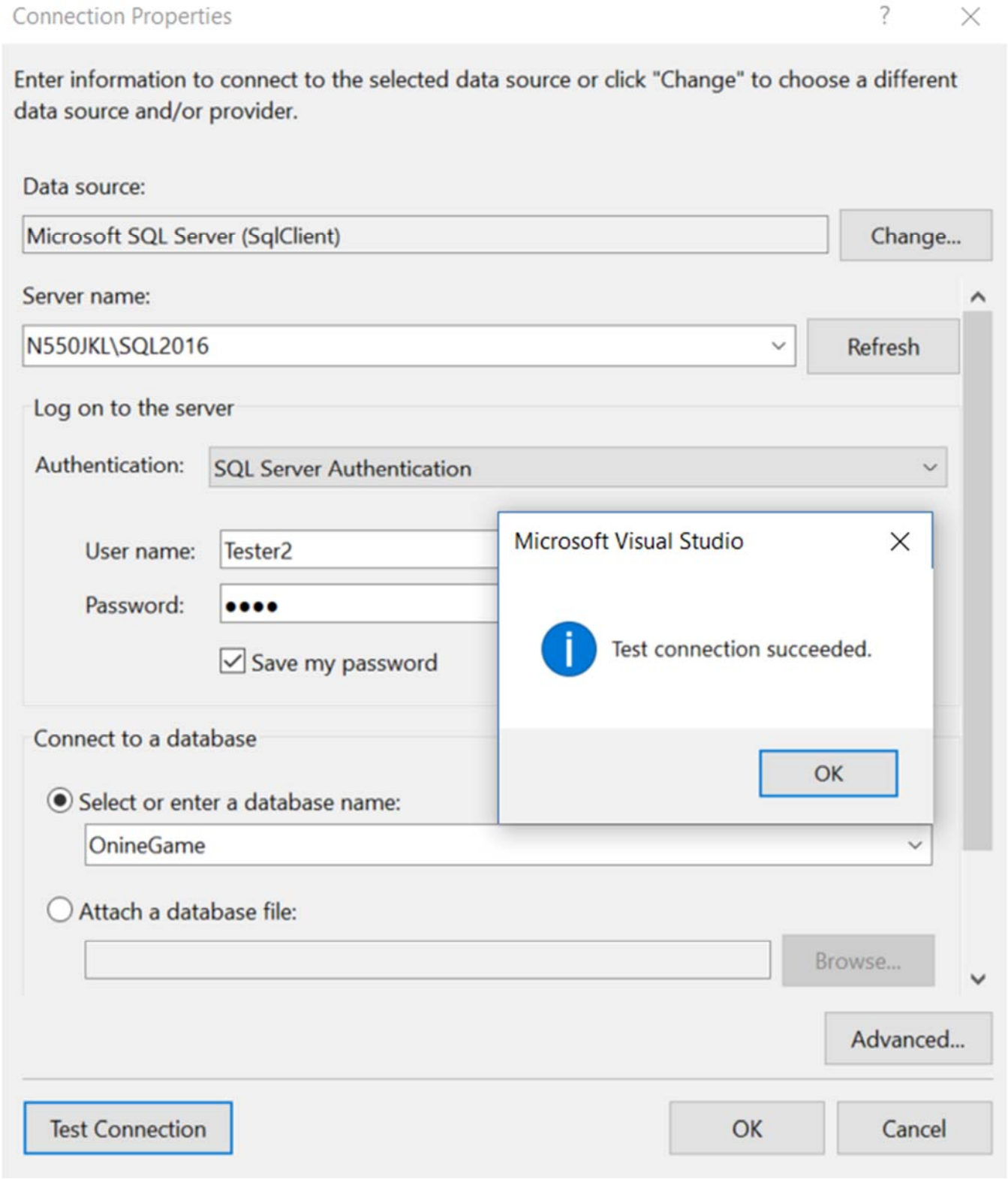
Save Connection settings in Web.Config as:

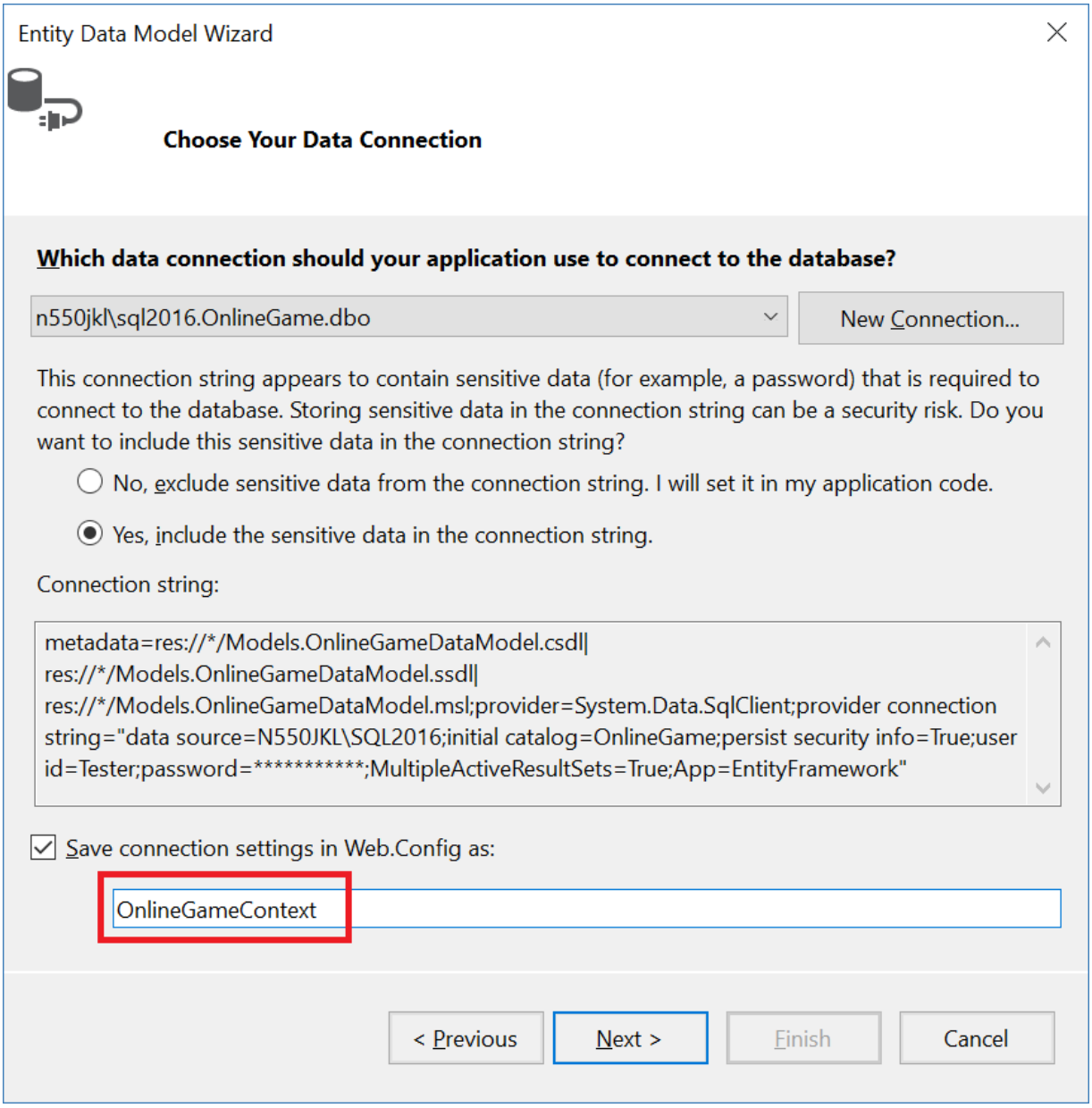
OnlineGameContext

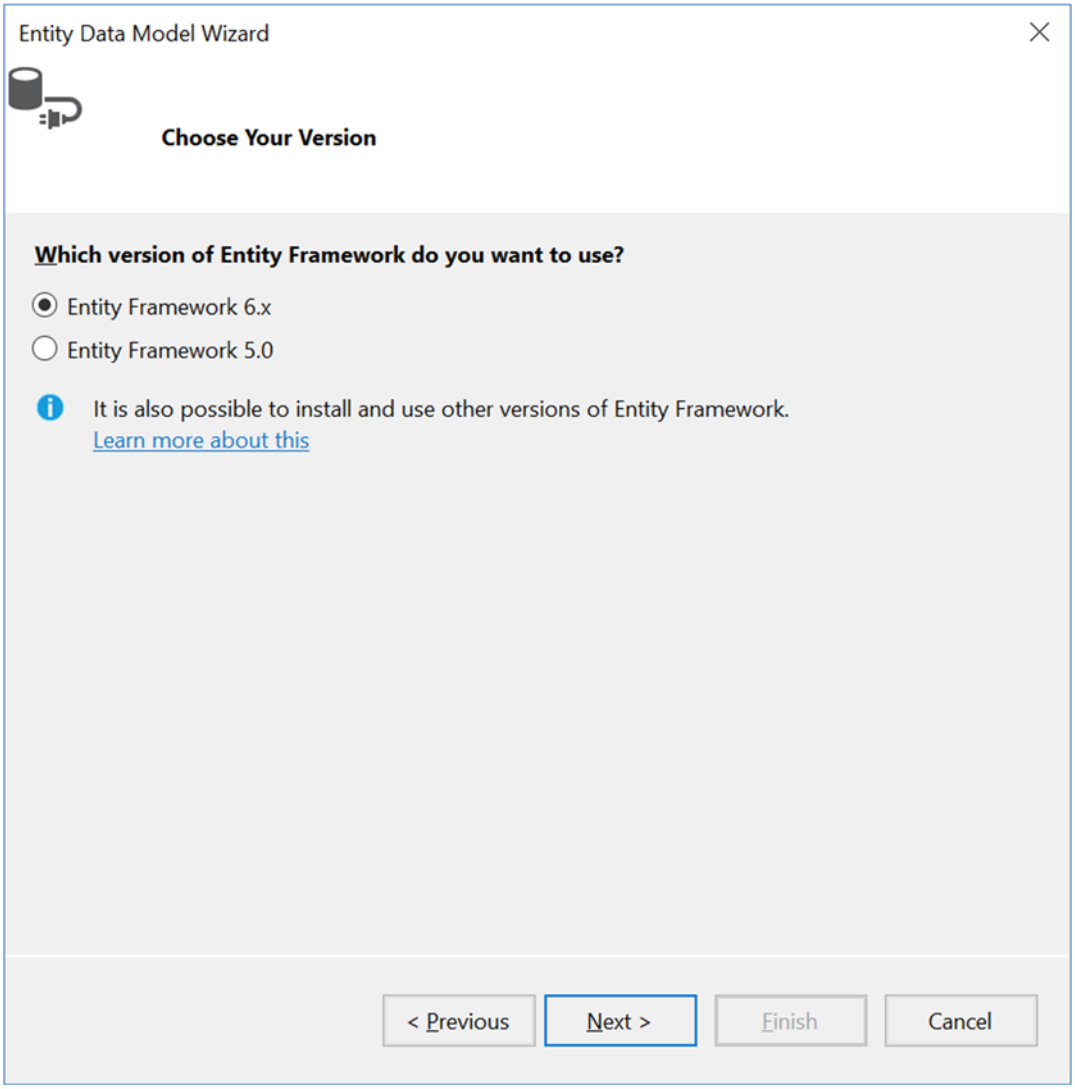


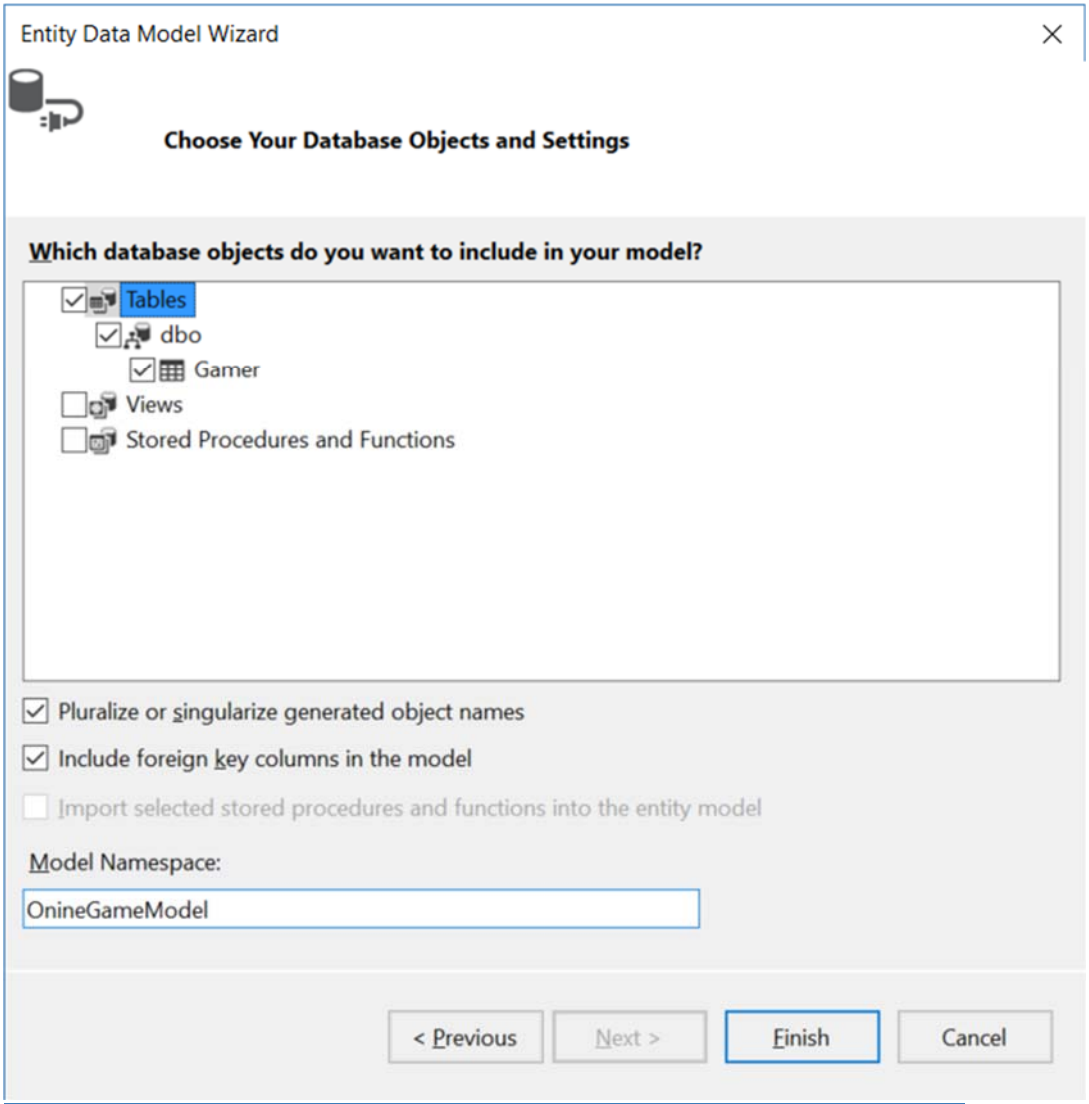


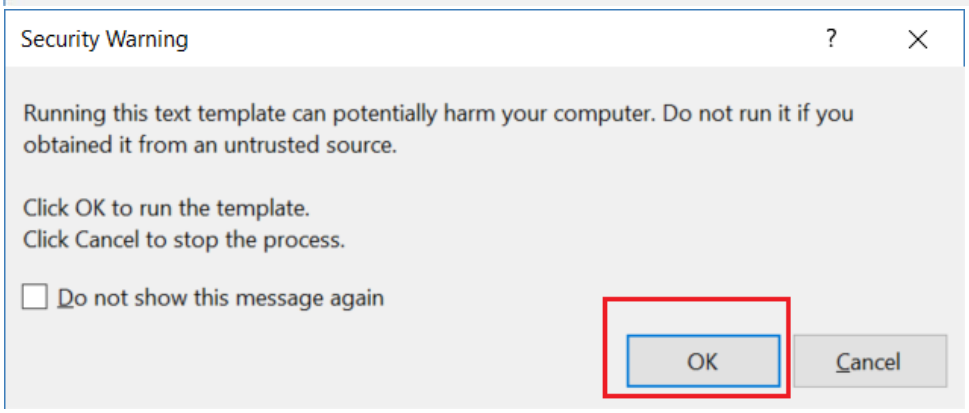

Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
New Connection...
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?
No, exclude sensitive data from the connection string. I will set it in my application code.
Yes, include the sensitive data in the connection string.
Connection string:
Save connection settings in Web.Config as:
< Previous
Next >
Finish
Cancel

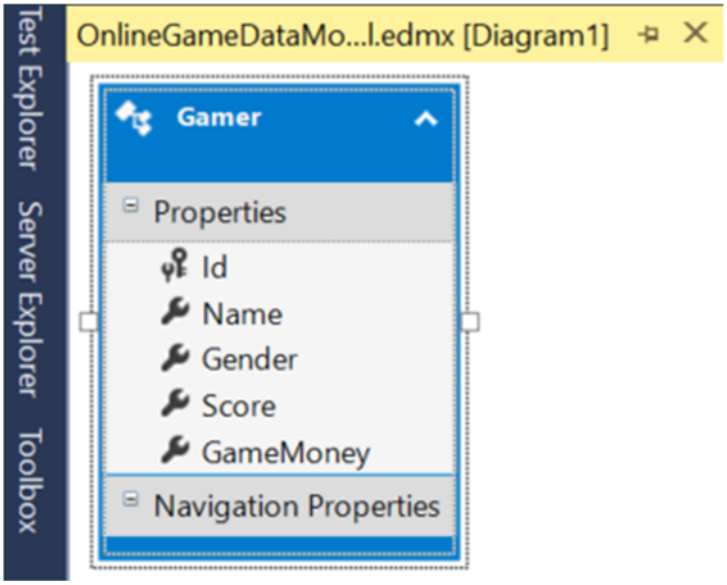

Connection Properties
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
N550JKL\SQL2016
Refresh
Log on to the server
Authentication:
SQL Server Authentication
User name:
Tester2
Password:
Save my password
Microsoft Visual Studio
Test connection succeeded.
OK
Connect to a database
Select or enter a database name:
OnineGame
Attach a database file:
Browse...
Advanced...
Test Connection
OK
Cancel


Entity Data Model Wizard
Choose Your Data Connection
Which data connection should your application use to connect to the database?
n550jkl\sql2016.OnlineGame.dbo
New Connection...
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?
No, exclude sensitive data from the connection string. I will set it in my application code.
Yes, include the sensitive data in the connection string.
Connection string:
metadata=res://*/Models.OnlineGameDataModel.csdl|
res://*/Models.OnlineGameDataModel.ssdl|
res://*/Models.OnlineGameDataModel.msl;provider=System.Data.SqlClient;provider connection
string="data source=N550JKL\SQL2016;initial catalog=OnlineGame;persist security info=True;user
id=Tester;password=**********;MultipleActiveResultSets=True;App=EntityFramework"
Save connection settings in Web.Config as:
OnlineGameContext
< Previous
Next >
Finish
Cancel


Entity Data Model Wizard
Choose Your Version
Which version of Entity Framework do you want to use?
Entity Framework 6.x
Entity Framework 5.0
It is also possible to install and use other versions of Entity Framework.
Learn more about this
< Previous
Next >
Finish
Cancel


Entity Data Model Wizard
Choose Your Database Objects and Settings
Which database objects do you want to include in your model?
Tables
dbo
Gamer
Views
Stored Procedures and Functions
Pluralize or singularize generated object names
Include foreign key columns in the model
Import selected stored procedures and functions into the entity model
Model Namespace:
OnineGameModel
< Previous
Next >
Finish
Cancel

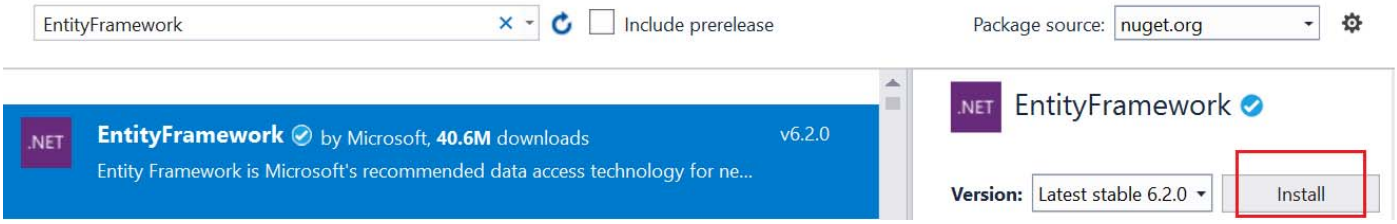

Security Warning
Running this text template can potentially harm your computer. Do not run it if you obtained it from an untrusted source.
Click OK to run the template.
Click Cancel to stop the process.
Do not show this message again
OK
Cancel



4. OnlineGame.WebApi

4.1. Install Entity Framework

Tools --> NuGet Package Manager --> Manage NuGet Packages for Solutions...
--> Browse tab --> Search : **EntityFramework**
--> Install it



4.2. Web.config : Add Connection String

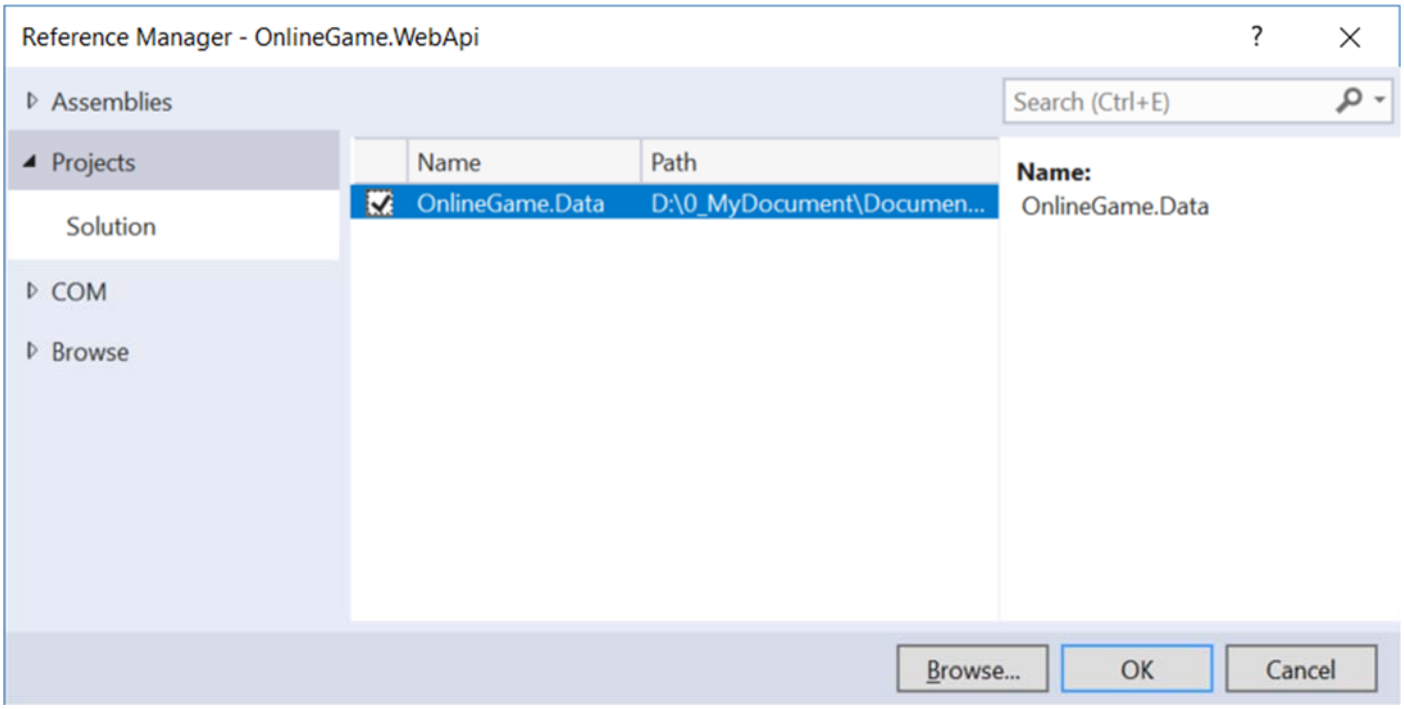
Web.config

```
69      </compilers>
70    </system.codedom>
71    <entityFramework>
72      <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
73        <parameters>
74          <parameter value="mssqllocaldb" />
75        </parameters>
76      </defaultConnectionFactory>
77      <providers>
78        <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices,
           EntityFramework.SqlServer" />
79      </providers>
80    </entityFramework>
81    <connectionStrings>
82      <add name="OnlineGameContext" connectionString="metadata=res://*/OnlineGameDataModel.csdl|res://*/
       OnlineGameDataModel.ssdl|res://*/OnlineGameDataModel.msl;provider=System.Data.SqlClient;provider connection
       string=&quot;data source=N550JKL\SQL2016;initial catalog=OnineGame;persist security info=True;user
       id=Tester2;password=1234;MultipleActiveResultSets=True;App=EntityFramework&quot;"
       providerName="System.Data.EntityClient" />
83    </connectionStrings>
84  </configuration>
```

```
<connectionStrings>
    <add name="OnlineGameContext" connectionString="metadata=res://*/OnlineGameDataModel.csdl|res://*/Onli
neGameDataModel.ssdl|res://*/OnlineGameDataModel.msl;provider=System.Data.SqlClient;provider connection
string=&quot;data source=N550JKL\SQL2016;initial catalog=OnineGame;persist security info=True;user
```

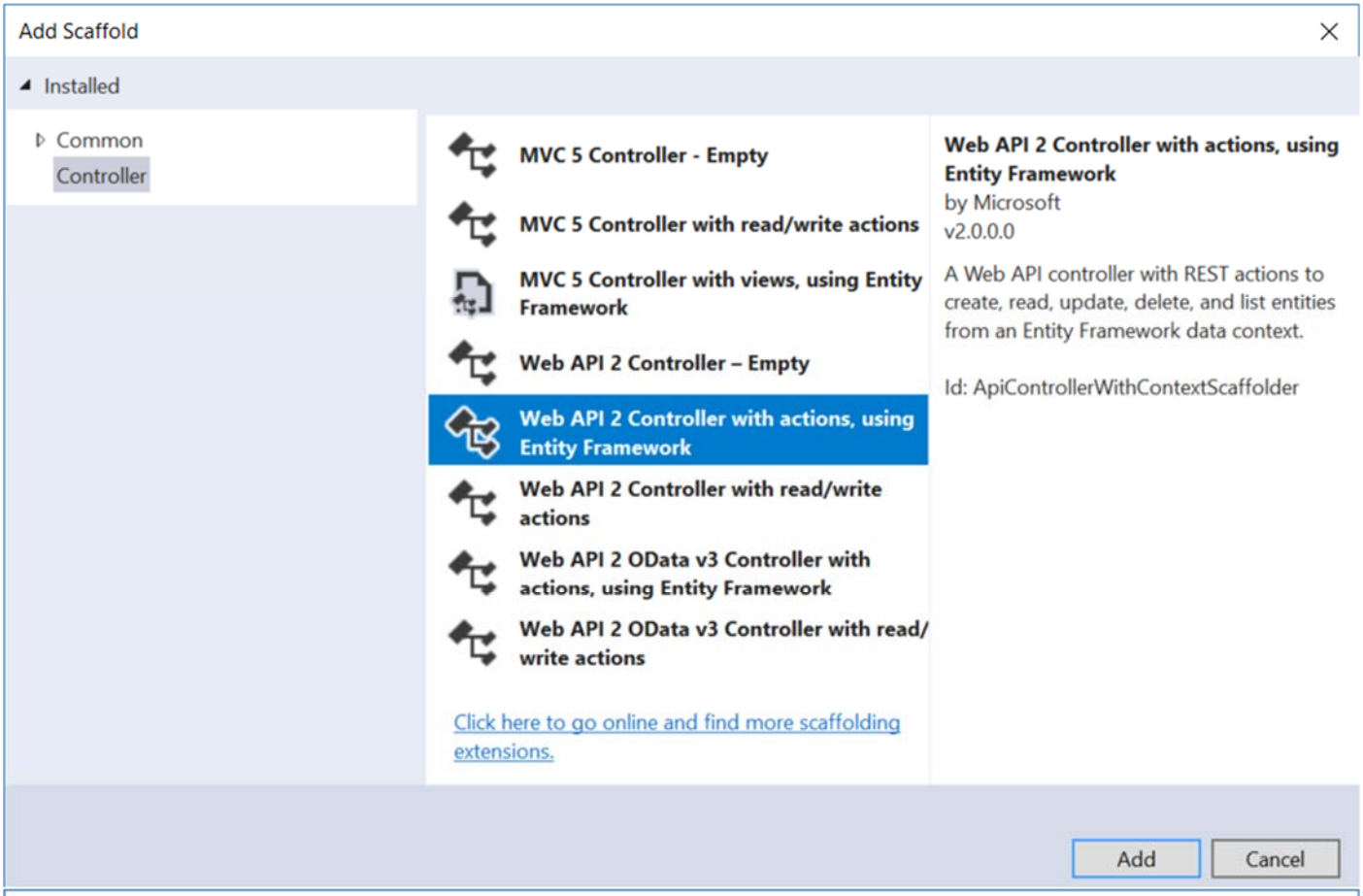
```
id=Tester2;password=1234;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />
  </connectionStrings>
```

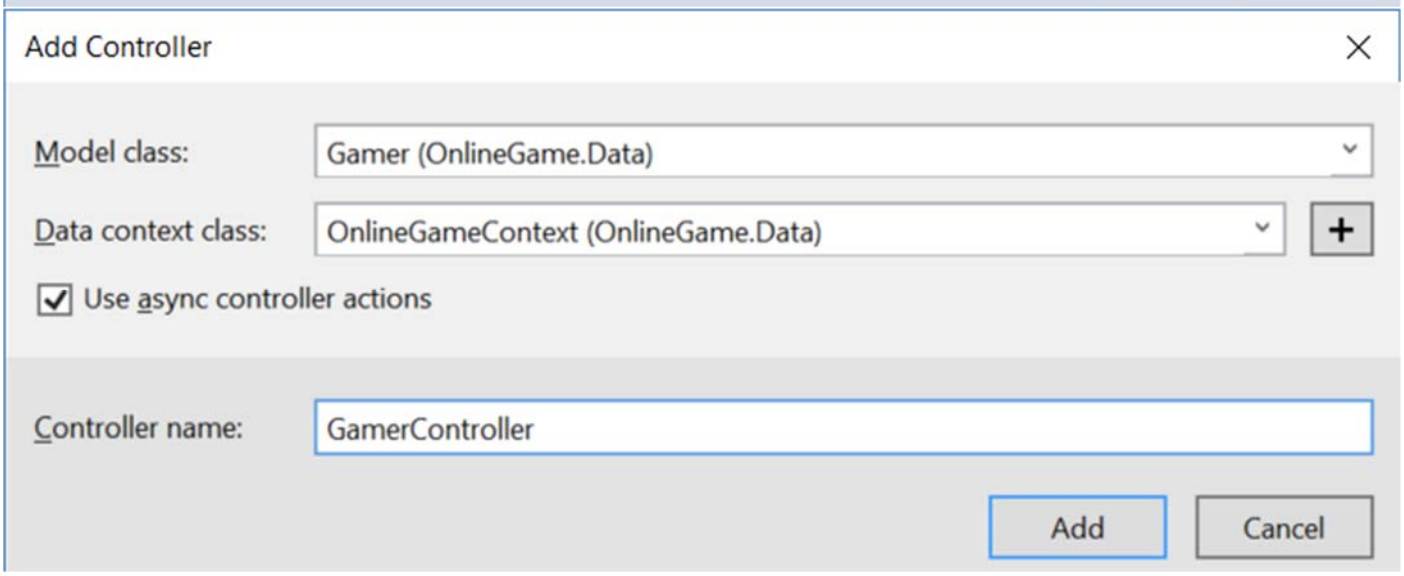
4.3. Add Reference



4.4. Controllers/GamerController.cs

Controllers folder --> Right Click --> Add --> Controller
--> **Web API 2 Controller with actions, using Entity Framework**
--> **GamerController**
if you have any error message, please ensure re-build whole solutions.





4.5. Controllers/GamerController.cs

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Threading.Tasks;
using System.Web.Http;
using System.Web.Http.Description;
using OnlineGame.Data;
```

```
namespace OnlineGame.WebApi.Controllers
{
    public class GamerController : ApiController
    {
        private OnlineGameContext _db = new OnlineGameContext();
        ////GET: api/Gamer
        //[HttpGet]
        //public IQueryable<Gamer> LoadGamers()
        ////public IQueryable<Gamer> GetGamers()
        //{
        //    return _db.Gamers;
        //}
        //GET: api/gamer?gender=female  --> Only Female Gamer
        //GET: api/gamer? gender = male-- > Only Male Gamer
        //GET: api/gamer --> All Gamers
        [HttpGet]
        public async Task<IHttpActionResult> LoadGamers(string gender = "")
        //public IQueryable<Gamer> GetGamers()
        {
            List<Gamer> gamers;
            switch (gender.ToLower())
            {
                case "male":
                    gamers = await _db.Gamers.Where(g => g.Gender.ToLower() == "male").ToListAsync();
                    break;
                case "female":
                    gamers = await _db.Gamers.Where(g => g.Gender.ToLower() == "female").ToListAsync();
                    break;
                default:
                    gamers = await _db.Gamers.ToListAsync();
                    break;
            }
            return Ok(gamers);   //200
        }
        // GET: api/Gamer/5
        [ResponseType(typeof(Gamer))]
        [HttpGet]
        public async Task<IHttpActionResult> LoadGamer(int id)
        //public async Task<IHttpActionResult> GetGamer(int id)
        {
            Gamer gamer = await _db.Gamers.FindAsync(id);
            if (gamer == null) return NotFound();  //404
            return Ok(gamer);   //200
        }
        // PUT: api/Gamer/5
        [ResponseType(typeof(void))]
        //public async Task<IHttpActionResult> PutGamer(int id, Gamer gamer)
        [HttpPut]
        //public async Task<IHttpActionResult> UpdateGamer(int id, Gamer gamer)
        public async Task<IHttpActionResult> UpdateGamer([FromUri]int id, [FromBody]Gamer gamer)    //By
Default
        //public async Task<IHttpActionResult> UpdateGamer([FromBody]int id, [FromUri]Gamer gamer)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);  //400
            }
```

```
    //if (id != gamer.Id)   return BadRequest();
    ////1.
     gamer.Id = id;
     _db.Entry(gamer).State = EntityState.Modified;  //update the gamer
    //2.
    //Gamer currentGamer = await _db.Gamers.FirstOrDefaultAsync(g => g.Id == id);
    //if (currentGamer == null) return NotFound();  //404
    //currentGamer.Name = gamer.Name;
    //currentGamer.Gender = gamer.Gender;
    //currentGamer.Score = gamer.Score;
    //currentGamer.GameMoney = gamer.GameMoney;
    try
     {
         await _db.SaveChangesAsync();
         return Ok();    //200
     }
    catch (DbUpdateConcurrencyException)
     {
         if (!GamerExists(id)) return NotFound();  //404
         throw;
     }
 }
// POST: api/Gamer
 [ResponseType(typeof(Gamer))]
 [HttpPost]
public async Task<IHttpActionResult> InsertGamer([FromBody]Gamer gamer)
//public async Task<IHttpActionResult> PostGamer([FromBody]Gamer gamer)
 {
    if (!ModelState.IsValid) return BadRequest(ModelState); //400
     _db.Gamers.Add(gamer);
    await _db.SaveChangesAsync();
    //Return Created/201.
    //1.
    return CreatedAtRoute("DefaultApi", new { id = gamer.Id }, gamer);    //Created/201
    ////Return Created/201.
    ////2.
    ////If you want to return HttpResponseMessage()
    ////2.
    ////Create a HttpResponseMessage with status code 201 Item Created.
    ////Pass the gamer into 2nd parameter as the created value.
    //HttpResponseMessage message =
    //    Request.CreateResponse(HttpStatusCode.Created, gamer);
    ////The Headers.Location should know the URI of the created item.
    //message.Headers.Location = new Uri(Request.RequestUri +
    //    gamer.Id.ToString());
    //return message;   //Created/201
    ////Return OK/200.
    ////3.
    ////if you want to return OK/200 when item created.
    //return Created(new Uri(Request.RequestUri + gamer.Id.ToString()), gamer);    //OK/200
 }
// DELETE: api/Gamer/5
 [ResponseType(typeof(Gamer))]
//[HttpDelete]
//public async Task<IHttpActionResult> RemoveGamer(Gamer gamer)
```

```
        public async Task<IHttpActionResult> DeleteGamer(int id)
        {
            Gamer gamer = await _db.Gamers.FindAsync(id);
            if (gamer == null) return NotFound();   //404
            _db.Gamers.Remove(gamer);
            await _db.SaveChangesAsync();
            return Ok(gamer);   //200
        }
        protected override void Dispose(bool disposing)
        {
            if (disposing) _db.Dispose();   //Dispose DBContext
            base.Dispose(disposing);
        }
        private bool GamerExists(int id)
        {
            return _db.Gamers.Count(e => e.Id == id) > 0;
        }
    }
}
/*
1.
1.1.
By default, the HTTP verb GET maps to a method that has the name Get() or "Get" prefix.
E.g. Get(), GetGamers, GetXXX()
If you want the HTTP verb GET maps to the method name without "Get" prefix.
You can use [HttpGet] attribute.
1.2.
[HttpGet] attribute maps HTTP verb GET.
[HttpPost] attribute maps HTTP verb POST.
[HttpPut] attribute maps HTTP verb PUT.
[HttpDelete] attribute maps HTTP verb DELETE.
----------------------------
2.
Web Api default binding parameter convention
2.1.
By default, if the parameter is a simple type,
Web Api will try to get value from uri.
E.g. int, double, bool, ...etc.
2.2.
By default, if the parameter is a complex type,
Web Api will try to get value from the request body.
E.g. Gamer
-----------------
2.3.
//[HttpPut]
//public async Task<IHttpActionResult> UpdateGamer(int id, Gamer gamer)
By Default, the Web Api will try to get id from uri, and gamer from request body as below code.
//[HttpPut]
//public async Task<IHttpActionResult> UpdateGamer([FromUri]int id, [FromBody]Gamer gamer)
E.g.
A.
PUT
http://localhost:58302/api/Gamer/8
B.
Request Header
Host: localhost:58302
Content-Type: application/json
B.1.
Accept: application/json
means we request JSON format response.
B.2.
Content-Type: application/json
The client will post a data to the server, the data format is JSON
C.
Request Body
```

```
{
"Name":"NameEight XYZ222",
"Gender":"Male",
"Score":450,
"GameMoney":1500
}
-----------------
2.4.
//[HttpPut]
//public async Task<IHttpActionResult> UpdateGamer([FromBody]int id, [FromUri]Gamer gamer)
[FromBody] will enfroce to get id from request body
[FromUri] will enforce to get gamer from uri
E.g.
A.
PUT
http://localhost:58302/api/Gamer?Name=NameEight%20XYZ333&Gender=Male&Score=450&GameMoney=1500
B.
Request Header
Host: localhost:58302
Content-Type: application/json
B.1.
Accept: application/json
means we request JSON format response.
B.2.
Content-Type: application/json
The client will post a data to the server, the data format is JSON
C.
Request Body
8
*/
```

E.g.1.1.

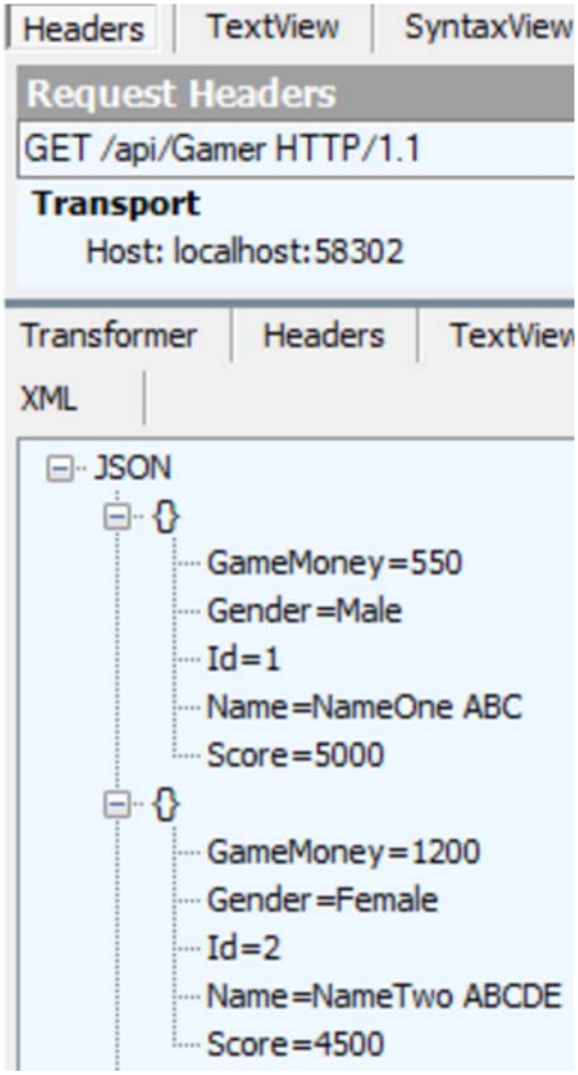
Get

http://localhost:58302/api/Gamer

Host: localhost:58302

-->

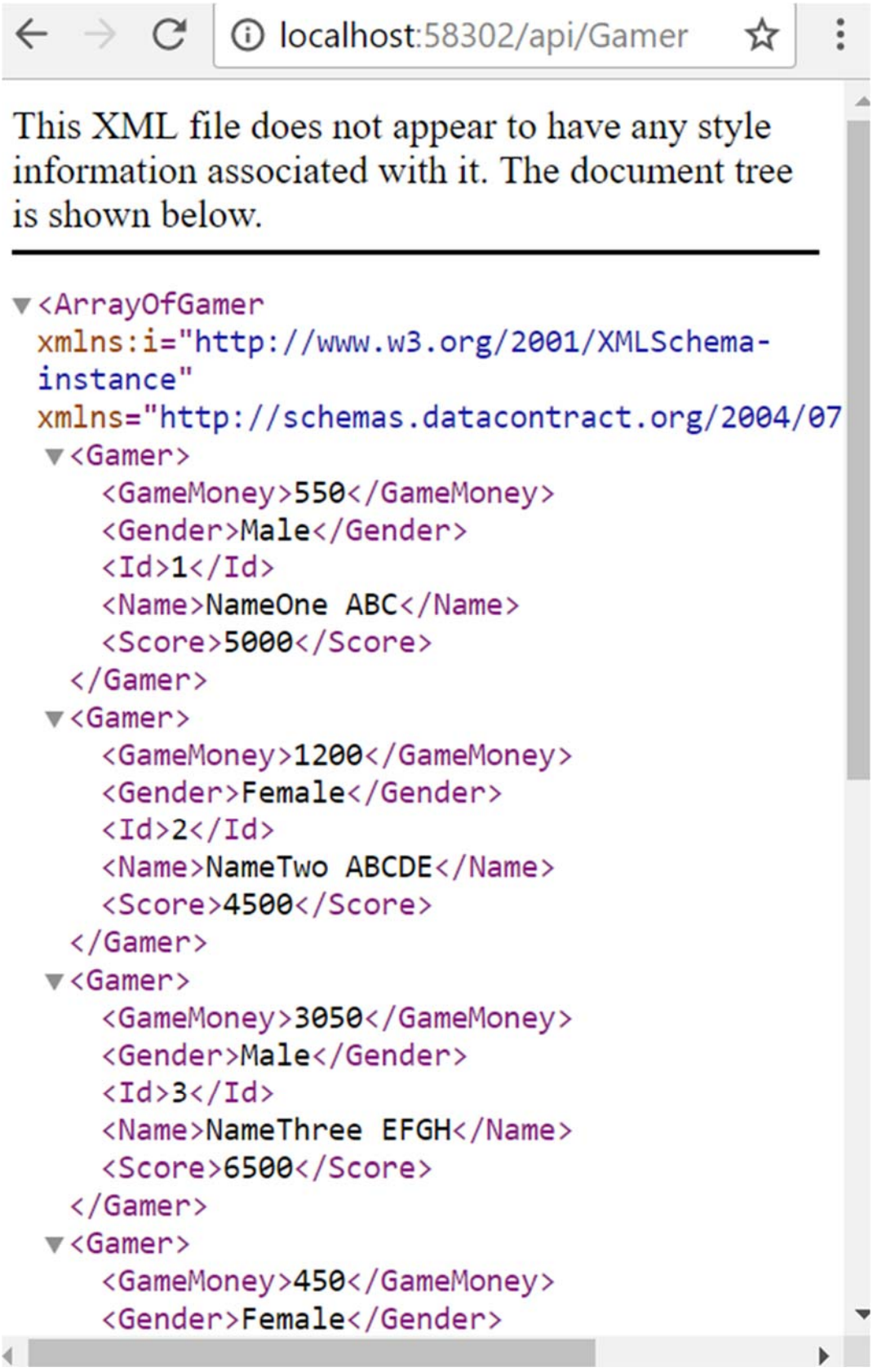
-->



E.g.1.2.

Get

http://localhost:58302/api/Gamer



E.g.1.3.

Get

http://localhost:58302/api/Gamer?gender=Female

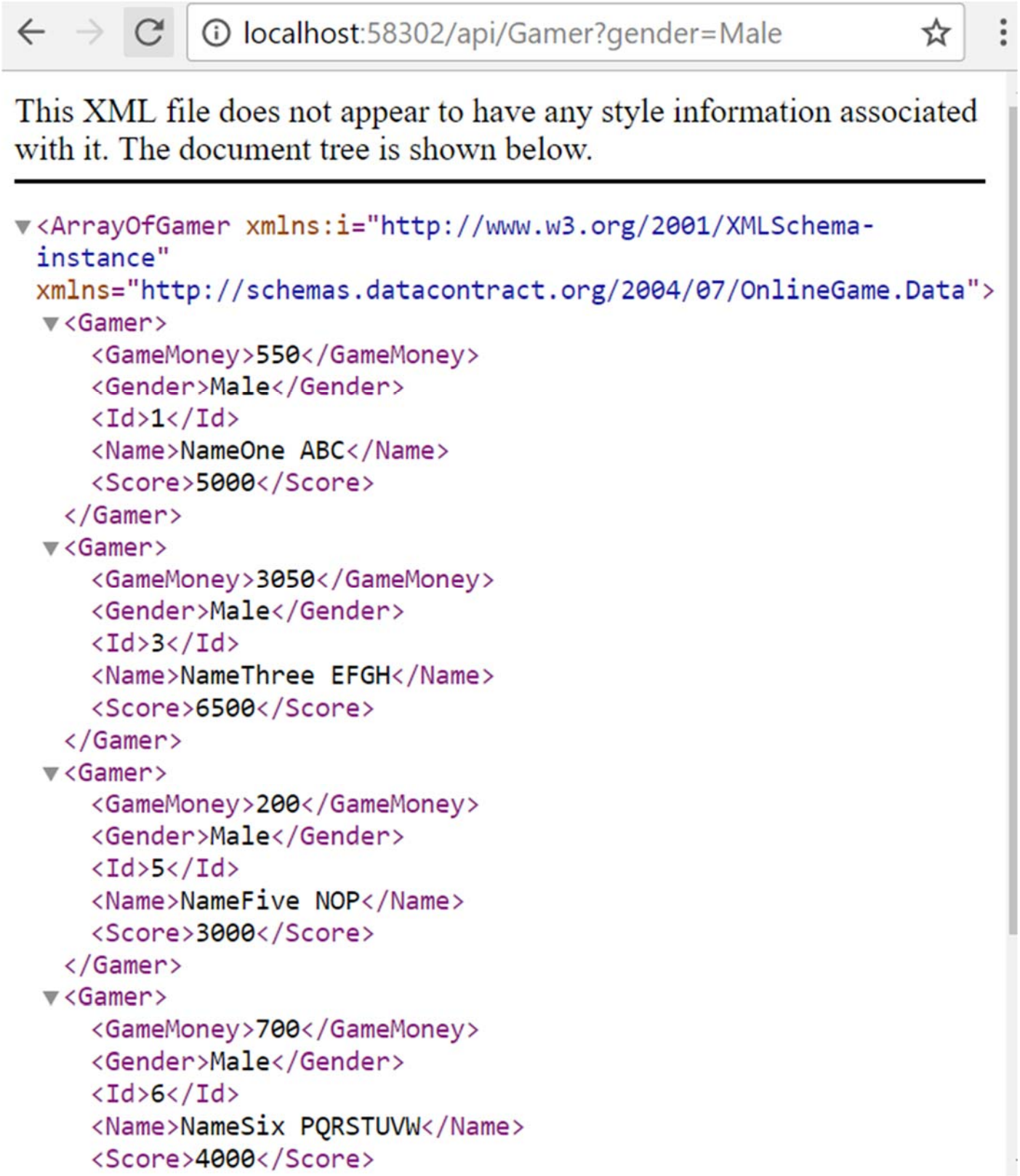
This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<ArrayOfGamer xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://schemas.datacontract.org/2004/07/OnlineGame.Data">
  ▼<Gamer>
     <GameMoney>1200</GameMoney>
     <Gender>Female</Gender>
     <Id>2</Id>
     <Name>NameTwo ABCDE</Name>
     <Score>4500</Score>
   </Gamer>
  ▼<Gamer>
     <GameMoney>450</GameMoney>
     <Gender>Female</Gender>
     <Id>4</Id>
     <Name>NameFour HIJKLMN</Name>
     <Score>45000</Score>
   </Gamer>
 </ArrayOfGamer>
```

E.g.1.4.

Get

http://localhost:58302/api/Gamer?gender=Male



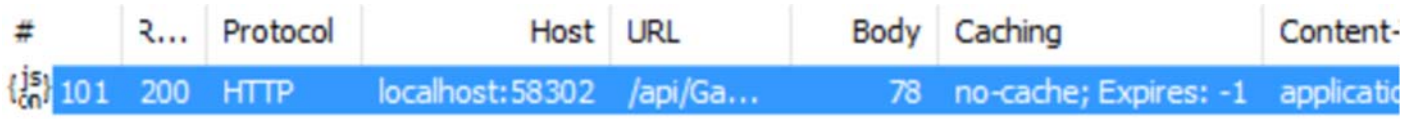
E.g.2.

Get(int id)

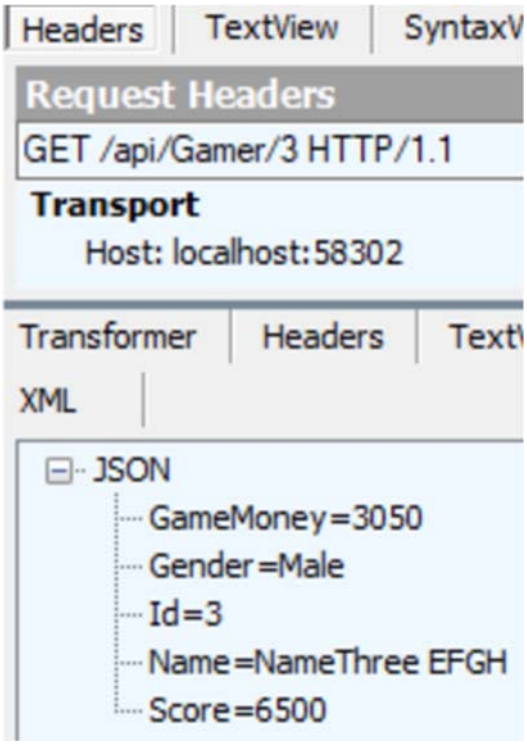
http://localhost:58302/api/Gamer/3

Host: localhost:58302

-->



-->



E.g.3.

A.

Post

http://localhost:58302/api/Gamer

B.

Request Header

Host: localhost:58302

Accept: application/json

Content-Type: application/json

B.1.

Accept: application/json

means we request JSON format response.

B.2.

Content-Type: application/json

Content-Length: 80

The client will post a data to the server, the data format is JSON and length is 80.

C.

Request Body

{

"Name":"NameEight XYZ",

"Gender":"Male",

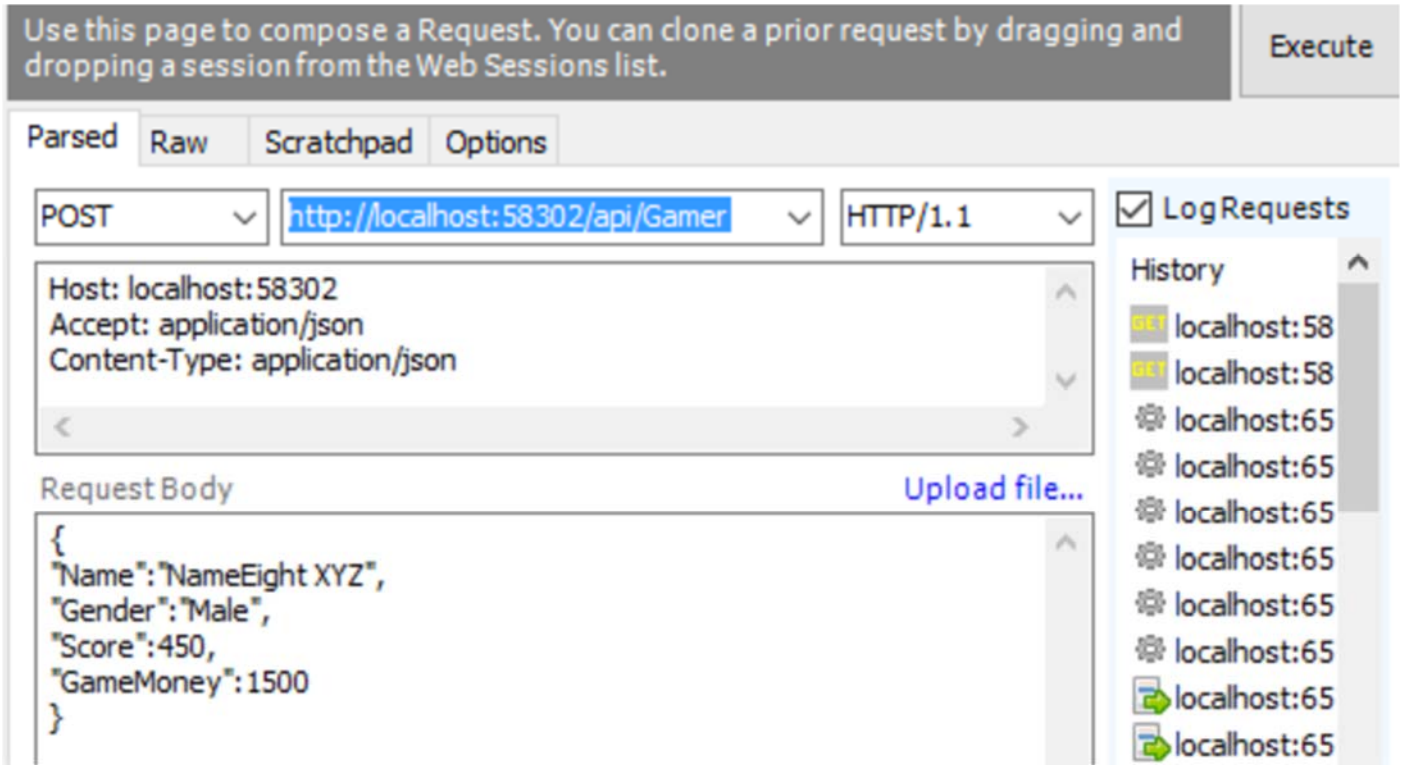
"Score":450,

"GameMoney":1500

}

	Id	Name	Gender	Score	GameMoney
1	1	NameOne ABC	Male	5000	550
2	2	NameTwo ABCDE	Female	4500	1200
3	3	NameThree EFGH	Male	6500	3050
4	4	NameFour HIJKLMN	Female	45000	450
5	5	NameFive NOP	Male	3000	200
6	6	NameSix PQRSTUVW	Male	4000	700
7	7	NameSeven XYZ	Male	450	1500

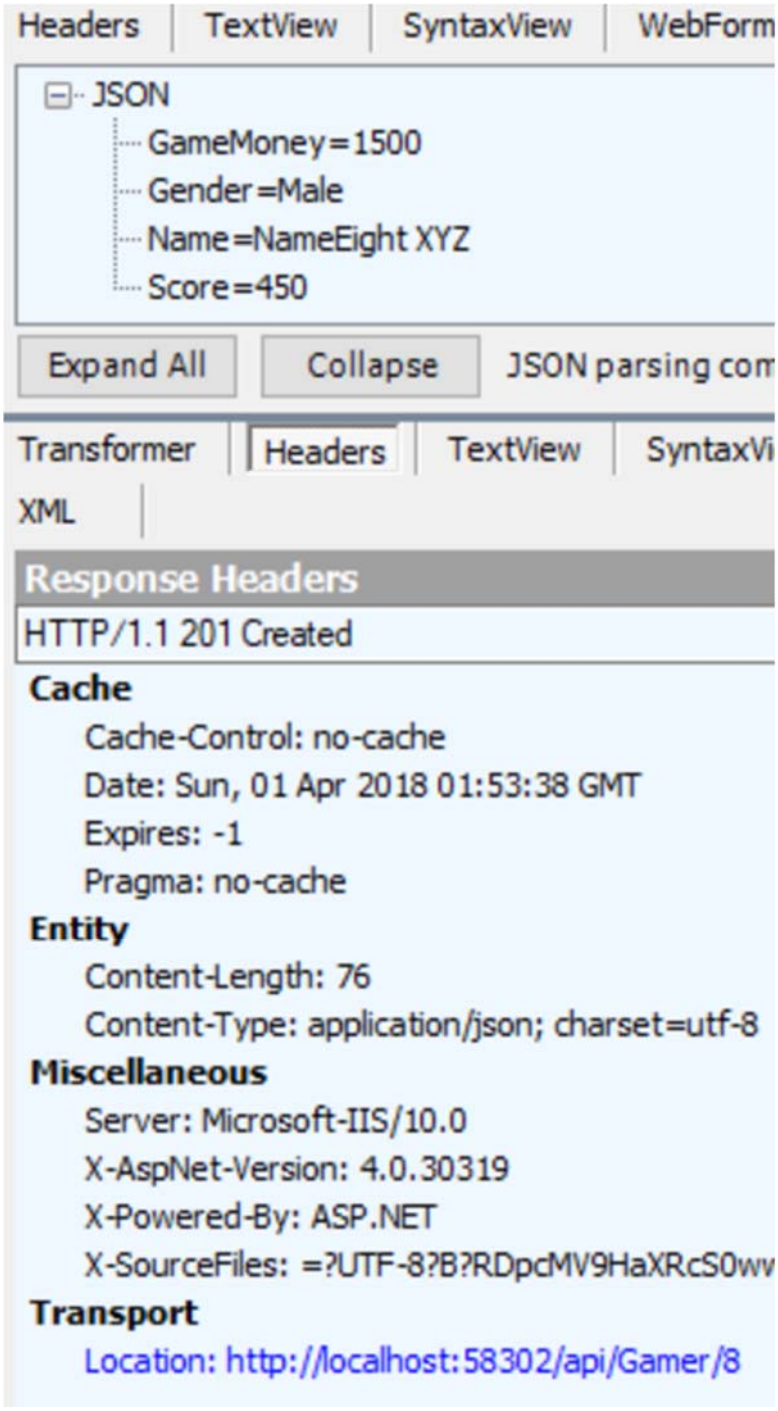
-->



-->

#	R...	Protocol	Host	URL	Body	Caching
102	201	HTTP	localhost:58302	/api/Ga...	76	no-cache; Expir

-->



-->

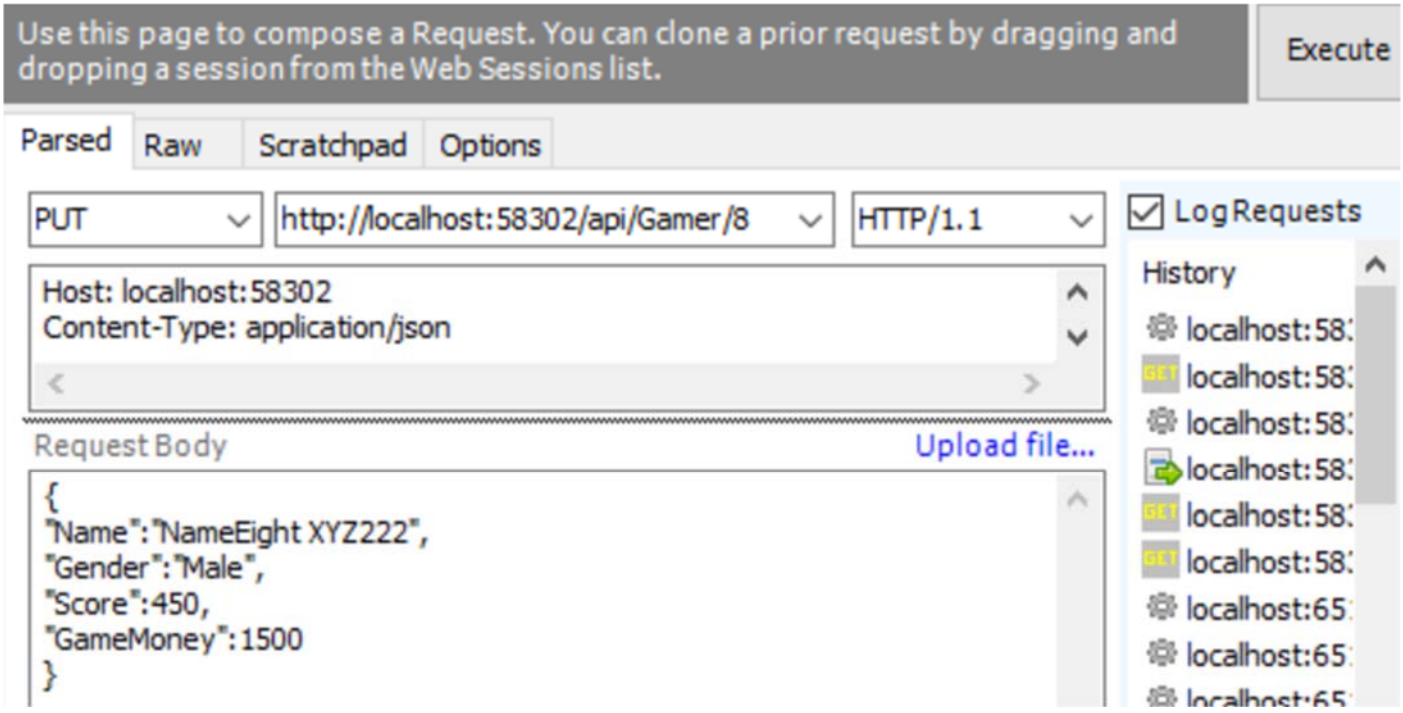
	Id	Name	Gender	Score	GameMoney
1	1	NameOne ABC	Male	5000	550
2	2	NameTwo ABCDE	Female	4500	1200
3	3	NameThree EFGH	Male	6500	3050
4	4	NameFour HIJKLMN	Female	45000	450
5	5	NameFive NOP	Male	3000	200
6	6	NameSix PQRSTUVW	Male	4000	700
7	7	NameSeven XYZ	Male	450	1500
8	8	NameEight XYZ	Male	450	1500

E.g.4.

A.
PUT
http://localhost:58302/api/Gamer/8
B.
Request Header
Host: localhost:58302
Content-Type: application/json
B.1.
Accept: application/json
means we request JSON format response.
B.2.
Content-Type: application/json
Content-Length: 80
The client will post a data to the server, the data format is JSON and length is 80.
C.
Request Body
{
"Name":"NameEight XYZ222",
"Gender":"Male",
"Score":450,
"GameMoney":1500
}

8	8	NameEight XYZ	Male	450	1500

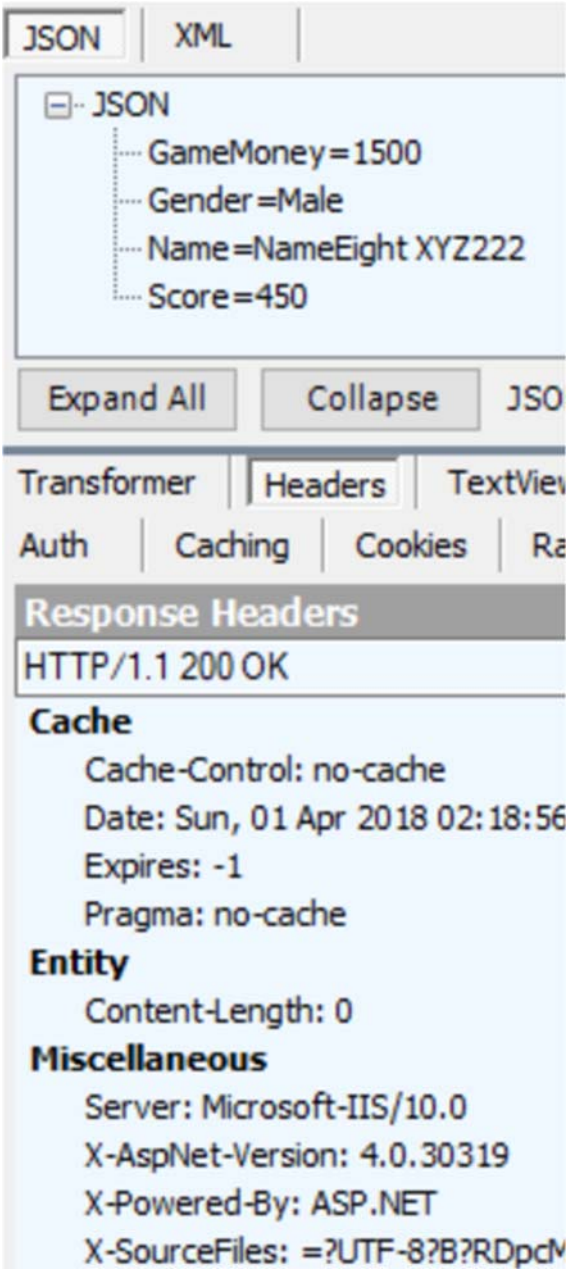
-->



-->

#	R...	Protocol	Host	URL	Body	Cachi
1	200	HTTP	localhost:58302	/api/Ga...	0	no-ca

-->



-->

8	8	NameEight XYZ222	Male	450	1500

A.

Delete

http://localhost:58302/api/Gamer/8

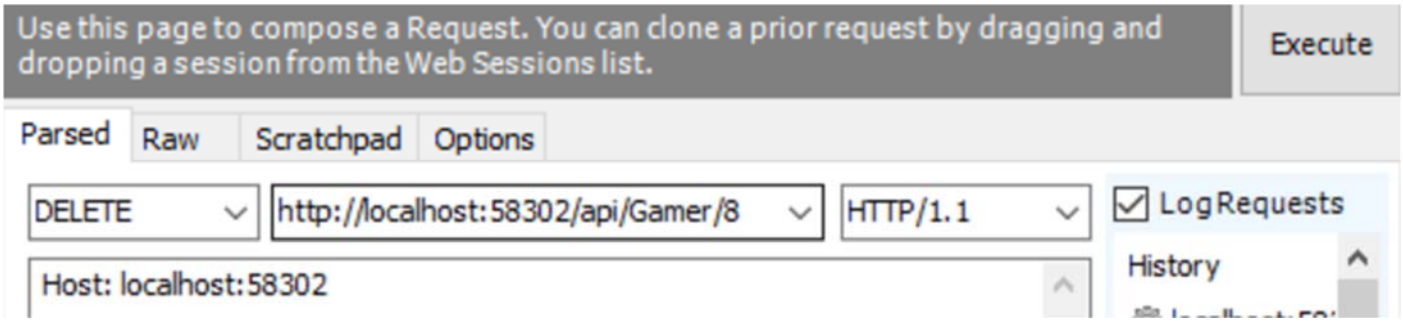
B.

Request Header

Host: localhost:58302

	Id	Name	Gender	Score	GameMoney
1	1	NameOne ABC	Male	5000	550
2	2	NameTwo ABCDE	Female	4500	1200
3	3	NameThree EFGH	Male	6500	3050
4	4	NameFour HIJKLMN	Female	45000	450
5	5	NameFive NOP	Male	3000	200
6	6	NameSix PQRSTUVW	Male	4000	700
7	7	NameSeven XYZ	Male	450	1500
8	8	NameEight XYZ222	Male	450	1500

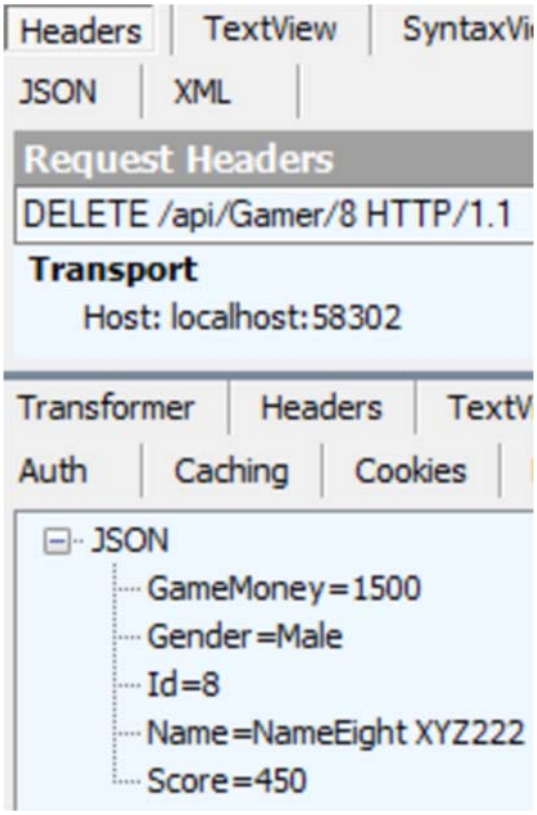
-->



-->

#	Result	Protocol	Host	URL	Body	Caching
1	200	HTTP	localhost:58302	/api/Ga...	79	no-cache; Expires: -1

-->



-->

	Id	Name	Gender	Score	GameMoney
1	1	NameOne ABC	Male	5000	550
2	2	NameTwo ABCDE	Female	4500	1200
3	3	NameThree EFGH	Male	6500	3050
4	4	NameFour HIJKLMN	Female	45000	450
5	5	NameFive NOP	Male	3000	200
6	6	NameSix PQRSTUVW	Male	4000	700
7	7	NameSeven XYZ	Male	450	1500

4.6. [FromBody] attribute and [FromUri] attribute

```
// PUT: api/Gamer/5
[ResponseType(typeof(void))]
[HttpPut]
//public async Task<IHttpActionResult> PutGamer(int id, Gamer gamer)
//public async Task<IHttpActionResult> UpdateGamer(int id, Gamer gamer)
//public async Task<IHttpActionResult> UpdateGamer([FromUri]int id, [FromBody]Gamer gamer)    //By
Default
public async Task<IHttpActionResult> UpdateGamer([FromBody]int id, [FromUri]Gamer gamer)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);  //400
    }
    //if (id != gamer.Id)   return BadRequest();
    ////1.
    gamer.Id = id;
    _db.Entry(gamer).State = EntityState.Modified;  //update the gamer
    ////2.
    //Gamer currentGamer = await _db.Gamers.FirstOrDefaultAsync(g => g.Id == id);
    //if(currentGamer == null) return NotFound();  //404
    //currentGamer.Name = gamer.Name;
    //currentGamer.Gender = gamer.Gender;
    //currentGamer.Score = gamer.Score;
    //currentGamer.GameMoney = gamer.GameMoney;
    try
    {
        await _db.SaveChangesAsync();
        return Ok();    //200
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!GamerExists(id)) return NotFound();  //404
        throw;
    }
}
```

2.
Web Api default binding parameter convention
2.1.
By default, if the parameter is simple type,
Web Api will try to get value from uri.

E.g. int, double, bool, ...etc.
2.2.
By default, if the parameter is complex type,
Web Api will try to get value from request body.
E.g. Gamer

2.3.
//[HttpPut]
//public async Task<IHttpActionResult> UpdateGamer(int id, Gamer gamer)
By Default, the Web Api will try to get id from uri, and gamer from request body as below code.
//[HttpPut]
//public async Task<IHttpActionResult> UpdateGamer([FromUri]int id, [FromBody]Gamer gamer)
E.g.
A.
PUT
http://localhost:58302/api/Gamer/8
B.
Request Header
Host: localhost:58302
Content-Type: application/json
B.1.
Accept: application/json
means we request JSON format response.
B.2.
Content-Type: application/json
The client will post a data to the server, the data format is JSON
C.
Request Body
{
"Name":"NameEight XYZ222",
"Gender":"Male",
"Score":450,
"GameMoney":1500
}

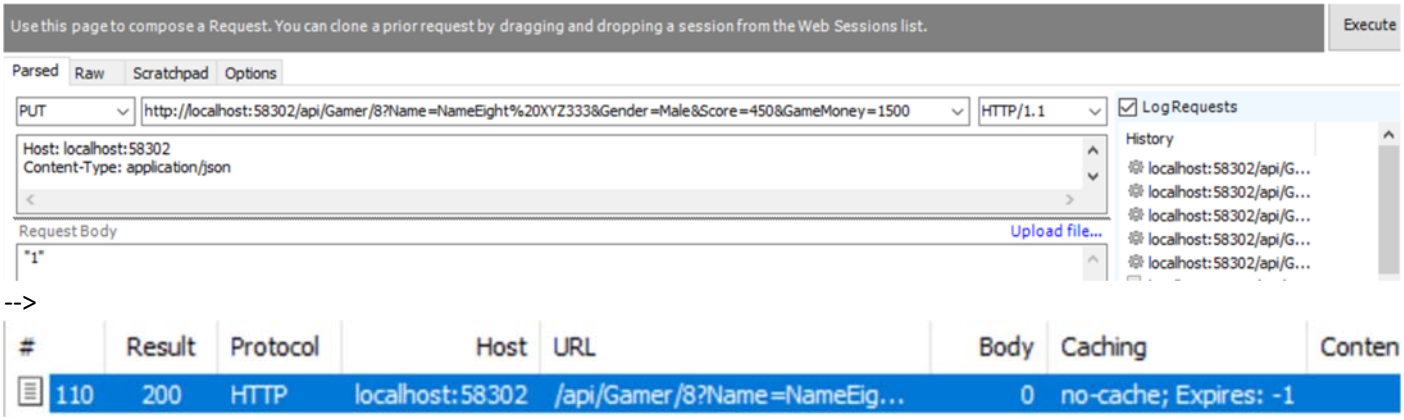
2.4.
//[HttpPut]
//public async Task<IHttpActionResult> UpdateGamer([FromBody]int id, [FromUri]Gamer gamer)
[FromBody] will enfroce to get id from request body
[FromUri] will enforce to get gamer from uri
E.g.
A.
PUT
http://localhost:58302/api/Gamer/8?Name=NameEight%20XYZ333&Gender=Male&Score=450&GameMoney=1500
B.
Request Header
Host: localhost:58302
Content-Type: application/json
B.1.
Accept: application/json
means we request JSON format response.
B.2.
Content-Type: application/json

The client will post a data to the server, the data format is JSON

C.

Request Body

"1"



-->