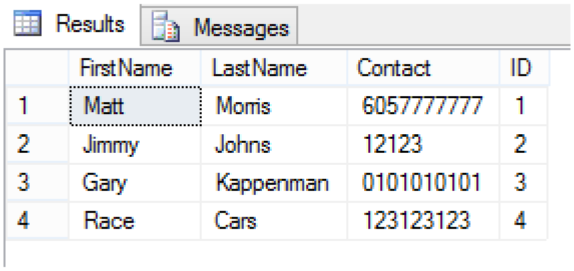
Visual Studio REST WebService that returns a JSON(Javascript Object Notation)

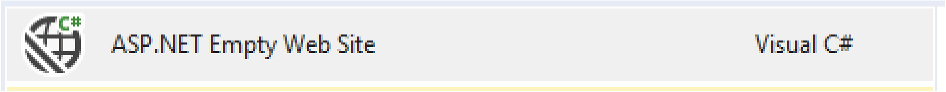
Description: Create REST web service using Visual Studio
Issues: Getting Visual Studio to behave properly with SQL server – I solved it by trouble shooting and stack overflow for a few hours. Changed "(LocalDB)\v11.0" to "(LocalDB)\MSSQLLocalDB"
Summary: After hours of trouble shooting finally got it to work. Assignment Complete.

1. Create SQL database containing "FirstName" ,"LastName" ,"Contact & "ID" fields.

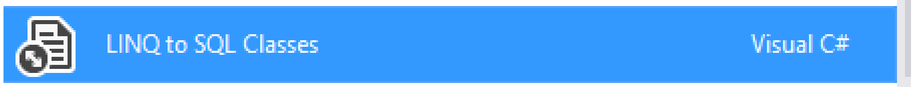


	FirstName	LastName	Contact	ID
1	Matt	Morris	6057777777	1
2	Jimmy	Johns	12123	2
3	Gary	Kappenman	0101010101	3
4	Race	Cars	123123123	4

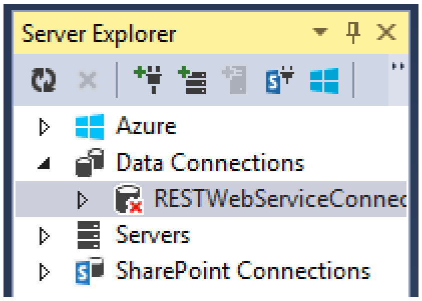
2. Open Visual Studio > File > New > Website > Empty Web Site > Finish

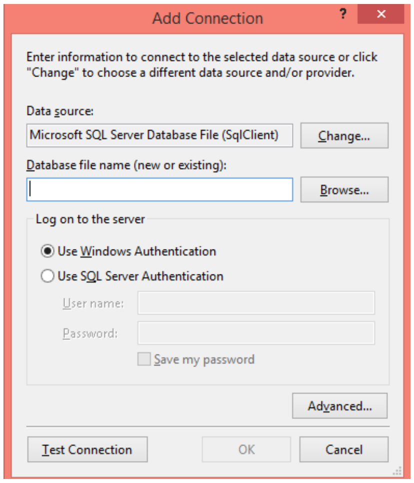


3. In solution explorer right click your project and click "Add new item"
4. Choose "LINQ to SQL Classes"



5. In server explorer(if not show, go to view > server explorer in menu) click "Add new connection", browse to your .MDF database in location like "C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\" then click "Test connection" and "OK"





Add Connection

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 Database File (SqlClient) Change...

Database file name (new or existing):
Browse...

Log on to the server

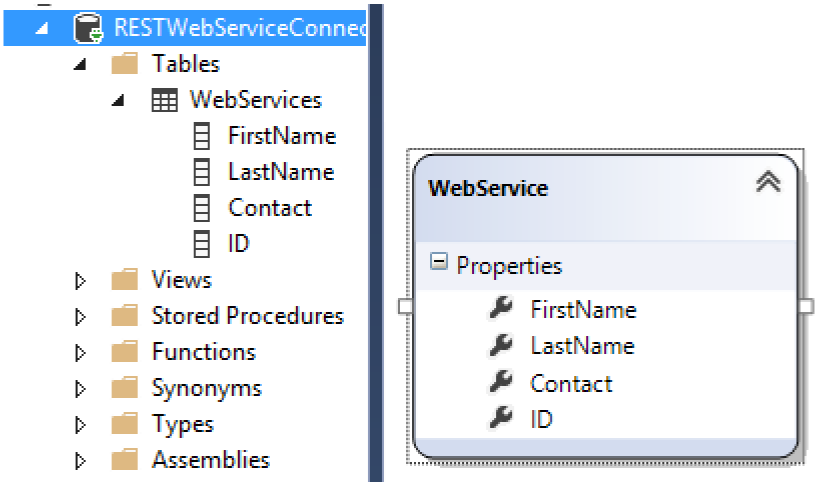
● Use Windows Authentication
○ Use SQL Server Authentication

User name:
Password:
□ Save my password

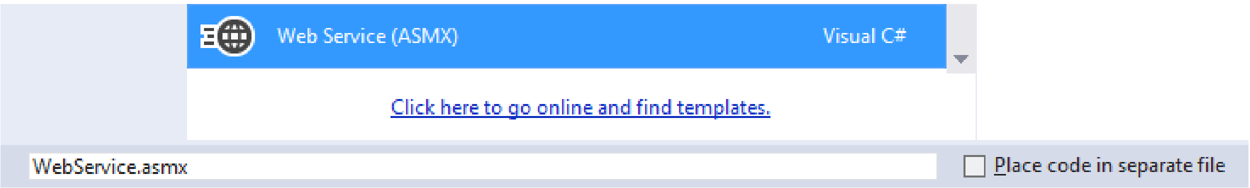
Advanced...

Test Connection OK Cancel

6. Next In server explorer open your database and drag your table onto your LINQ class canvas default named “DataClasses.dbml’ – you should now see your properties on the canvas.



7. In solution explorer right click your project and click “Add new Item”
8. Choose “Web Service” & UN-CHECK ‘Place code in separate file’ on your bottom right then click “Ok”


Web Service (ASMX)
Visual C#
Click here to go online and find templates.
WebService.asmx
□ Place code in separate file

9. Open your WebService.asmx and add these system statements listed below.

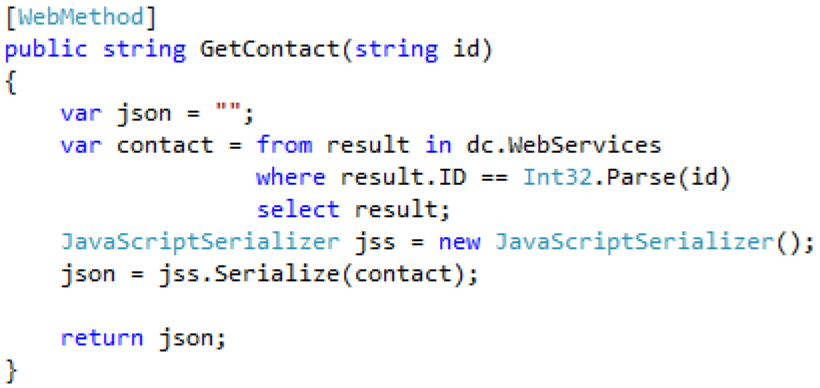
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Script.Serialization;
using System.Web.Services;
```

10. Create our data connection by typing “DataClassesDataContext dc = new DataclassesDataContext”

```
public class WebService2  : System.Web.Services.WebService {

    DataClassesDataContext dc = new DataClassesDataContext();
```

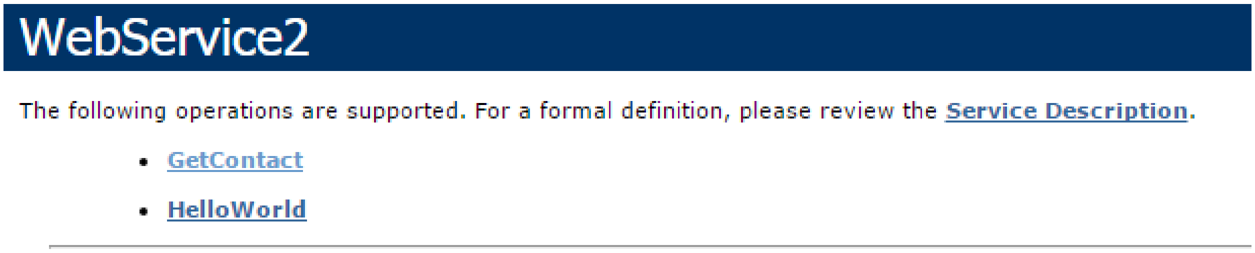
11. Now create a new method that returns our JSON which implements our SQL query that looks like the picture below.



```
[WebMethod]
public string GetContact(string id)
{
    var json = "";
    var contact = from result in dc.WebServices
                  where result.ID == Int32.Parse(id)
                  select result;
    JavaScriptSerializer jss = new JavaScriptSerializer();
    json = jss.Serialize(contact);

    return json;
}
```

12. Once complete, run your WebService.asmx and you'll see by default "HelloWorld" method and your new "GetContact" method.

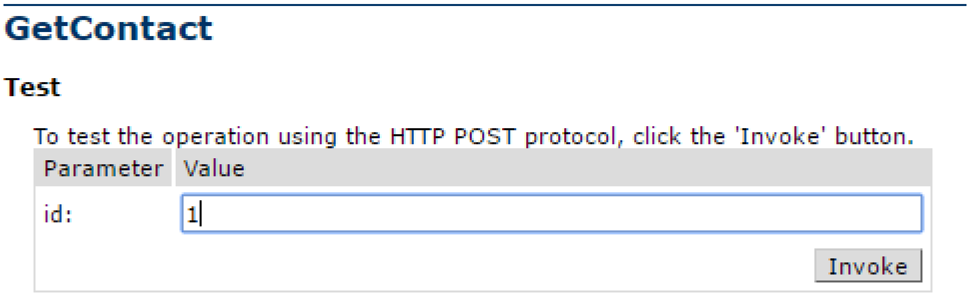


WebService2

The following operations are supported. For a formal definition, please review the Service Description.

- GetContact
- HelloWorld

13. Invoke your "GetContact" method and provide a parameter that you used in your ID SQL server table.



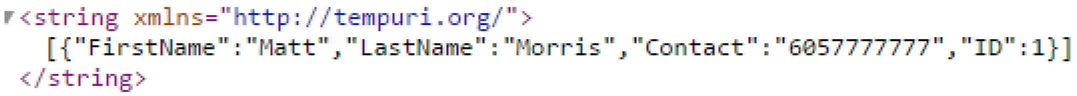
GetContact

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
id:	1

14. Your results are returned and you've created your Visual Studio RESTful Webservice ☺



```
▾<string xmlns="http://tempuri.org/">
   [{"FirstName":"Matt","LastName":"Morris","Contact":"6057777777","ID":1}]
 </string>
```