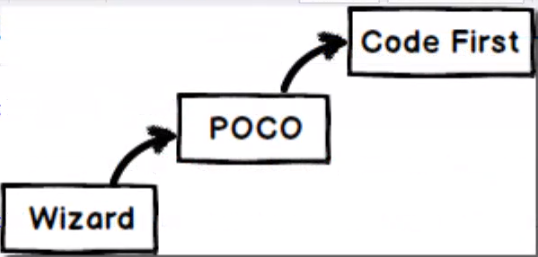
Integrating Entity with MVC

There are three method to connect and access with data layers.

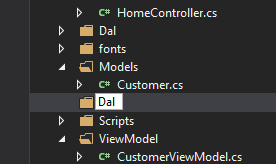
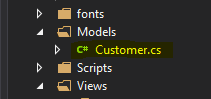


By Wizard Method it creates code automatically.

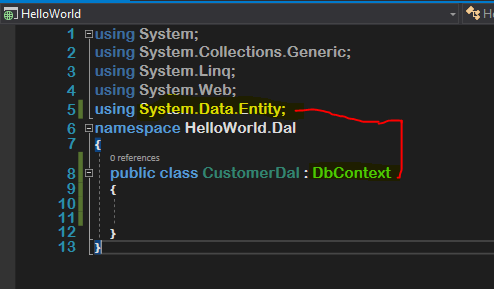
By POCO Method some of the code we have to add manually.

But Code Method we do everything. Here in this article we talk about Code First Method.

Add folder Dal in Solution explorer as under. And create class CustomerDal.cs

Edit and base class DbContext as under



Here CustomerDal class goes to Data base of Customer and fetch data from database and give us Customer Object. For that I need call Customer object for reference as under.

using HelloWorld.Models;

And do follows

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data.Entity;

using HelloWorld.Models;

namespace HelloWorld.Dal

{

    public class CustomerDal : DbContext

    {

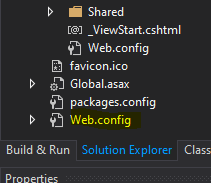
        public DbSet<Customer> Customers { get; set; }

    }

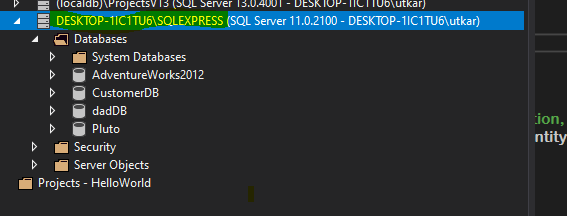
}

Now how does this Customer Data Access Layer know that where your data is located in SQL Server ? For that We need Connection string.

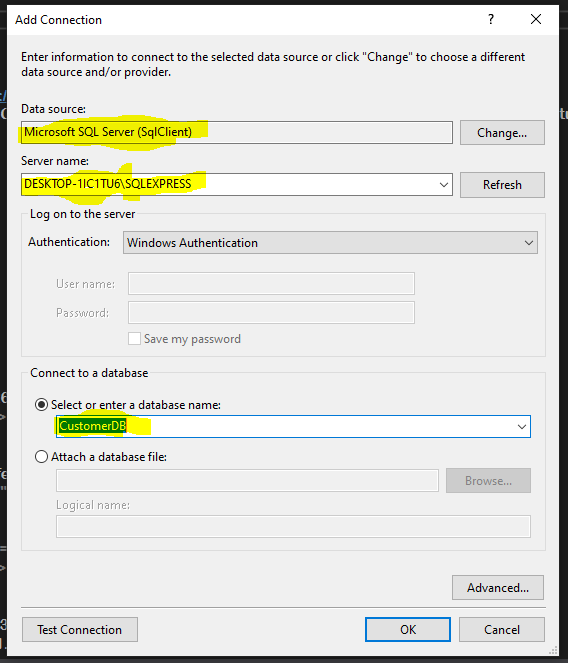
Go AS follows to file Web.config as under.

 Edit as follow.

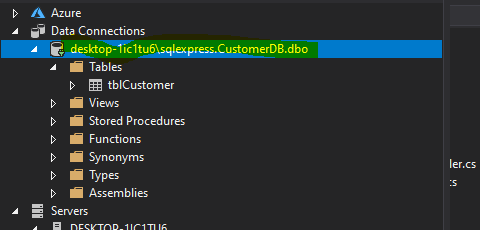
Find Server Name DESKTOP-1IC1TU6\SQLEXPRESS

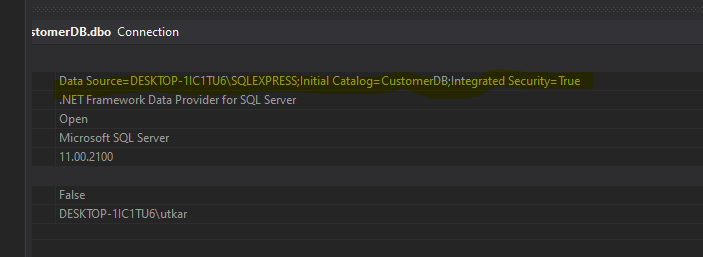


And add in Add Connection to find drop down menu of data base list in Select or enter Database name.

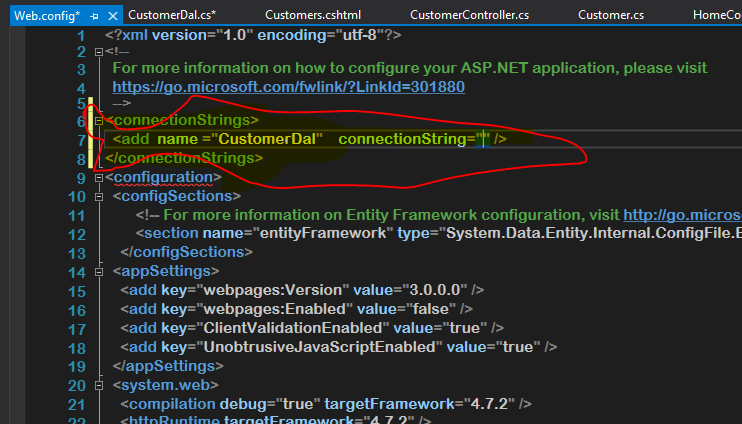


Find Property

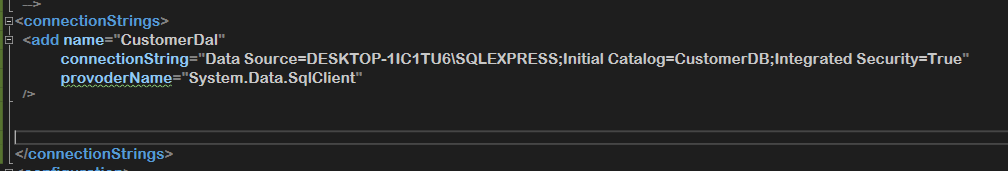




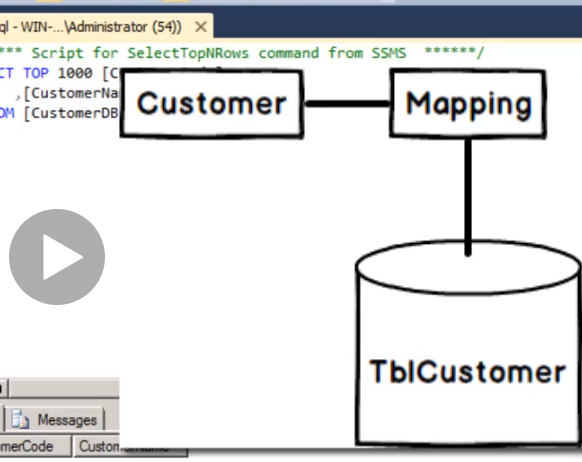
.NET Framework Data Provider for SQL Server



Add Provider Name



Need Mapping as database name is tblCustomer and entity name is Customer



Change CustomerDal.cs as under for mapping

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data.Entity;

using HelloWorld.Models;

namespace HelloWorld.Dal

{

    public class CustomerDal : DbContext

    {

        protected override void **OnModelCreating**(DbModelBuilder **modelBuilder**)

        {

            base.**OnModelCreating**(**modelBuilder**);

**modelBuilder**.**Entity**<Customer>().**ToTable**("tblCustomer");

        }

        public DbSet<Customer> Customers { get; set; }

    }

}

Now for saving go to CustomerController.cs and inport

using HelloWorld.Dal;

Final change in Submit action

        public ActionResult **Submit**( Customer **obj**)

        {

            /\*Customer ccc = new Customer();

            ccc.CustName = CustName;

            ccc.CustId = CustId;

            var name = CustName;

            var no = CustId;\*/

            var **name** = **obj**.CustName;

            var **no** = **obj**.CustId;

            ViewData["C1"] = **name**;

            ViewData["C2"] = **no**;

**if** (ModelState.IsValid)

            {

                //Insert Customer Objsct to data baseEF DAL

                CustomerDal **Dal** = new CustomerDal();

**Dal**.Customers.**Add**(**obj**);     //in memmory

**Dal**.**SaveChanges**();    //Physical  Commit

                ViewData["SUC"] = "Successfull";

**return** **View**("Customers", **obj**);

            }

            else

            {

**return** **View**("Customers", **obj**);

            }

            //return View("Customers",obj);

        }

    }

}

**Error:**  
"Only one <configSections> element allowed. It must be the first child element of the root <configuration> element".  
  
If you read the error carefully, it states that only one <configSections> element is allowed inside the Web.config and it should be the first child element and placed at the top. The reason for the error is that I accidentally placed the <connectionStrings></connectionStrings> at the top over the <configSections></configSections> and by conventions this is a violation. So, to fix the error, I rearranged the elements and the error was fixed

