OnlineBUFF

Home MSBI DOT.NET PHP Real Estate C# Interview Videos

Home DOT.NET Interview Questions Step by step 3 tier architecture in asp.net using c# example

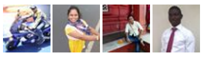
Three tier architecture in asp.net using c# example

Author: Gurunatha Dogi May 22nd, 2014 **Comments : 42** | Views : **144729**

In this tutorial post we will demonstrate how to create 3 tier architecture using asp.net c#. Three tier architecture means dividing our project into three layers that is presentation layer (UI layer), Business Layer (Logic code layer) and datalayer (Layer which connects to database).

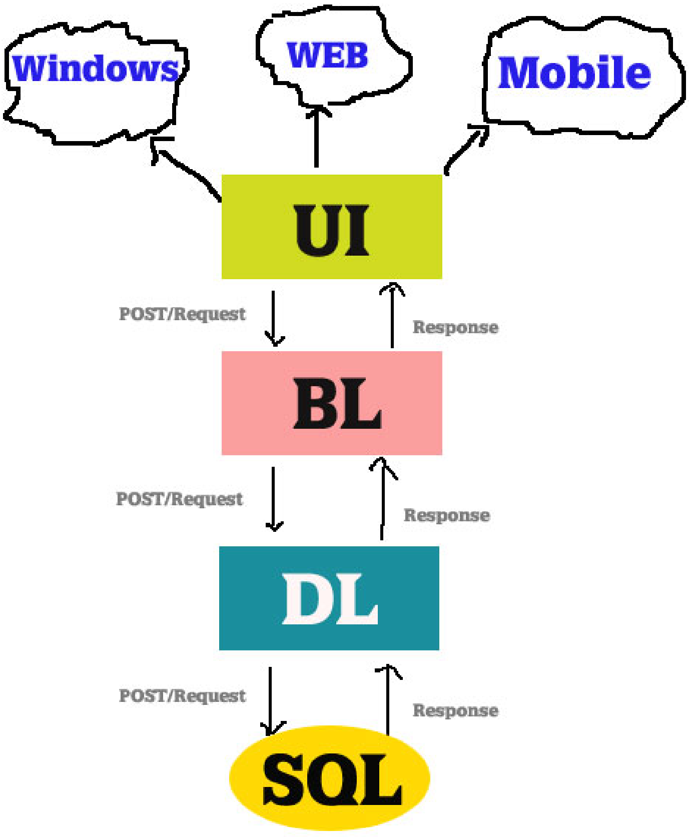
Many developers via emails ask me, how to implement three tier architecture in their application. There are lots of mails coming every day now and then asking for 3 tier architecture implementation. So this post is especially for those developers who are looking to develop a 3 tier implementation in their projects. Three tier architecture means dividing project in three layers User Interface Layer, Business Layer and Data Layer where we separate logic,data and user interface in three division. It means tomorrow if we replace sql server database with oracle database then we need to change only data layer OR if we change or user screen from windows application to website then we need to change only our user interface layer rest everything remain the same.

Kindly Note : Developer or coder need to be well versed with use of classes and objects in c-sharp.



Categories

MSBI Tutorials
DOT.NET Interview Questions
PHP Interview Questions
Real Estate
C# Interview Questions



Structure of 3 tier Architecture

How it works

Presentation layer is your user interface layer where you can design your interface using html web controls or windows controls or mobile controls. It can be your website or windows application or mobile application.This layer only communicates with business layer.

Application layer or Business Layer is the middle layer or bridge layer that connects database layer and presentation layer. In this layer you can write your business logic (logic written as per business requirement) or any validation code.This layer communicates with database layer and presentation layer.

Database layer which makes the connection to the database server. In this layer you can write database connection and sql queries or stored procedures. This layer only communicates with business layer.

When user post any data from from user interface (presentation page), data first goes to the business layer there data will be validated then validated data is posted to database layer to insert into the database.

When user request any data from the database then the request is first processed by the business layer validates the request and sends to database layer then database layer forwards it to the database server and fetches necessary records. Found records are loaded by the database layer and passes it back to the business layer then business layer passes those records to the presentation layer.

Advantages of using 3 tier architecture

It makes the logical separation between business layer and presentation layer and database layer.

Migration to new graphical environments is faster.

As each tier is independent it is possible to enable parallel development of each tier by using different sets of developers.

Easy to maintain and understand large project and complex project.

Since application layer is between the database layer and presentation layer so the database layer will be more secured and client will not have direct access to the database.

Posted data from presentation layer can be verified or validated at application layer before updating it to the database.

Database Security can be provided at application layer.

Application layer or middle layer or business layer can be a protection shield to the database.

New rules or new validation rules can be defined any time and changes made to middle layer will not effect presentation layer.

Define any logic once within the business layer and that logic can be shared among any number of components in the presentation layer.

We can display only necessary methods from business layer in the presentation layer.

We can hide unnecessary methods from business layer in the presentation layer.

Easy to apply object oriented concept

Easy to update data provider queries.

Disadvantage of using 3 tier architecture

Time consuming : To implement even small part of application it will consume lots of time.

Need good expertise in object oriented concept (classes and objects).

It is more complex to build.

Why to use 3 tier architecture in projects

To Manage Large Projects becomes difficult : Lets suppose you have created a 1000 screen project without using 3 tier architecture and coded all codes in the presentation layer then suppose over period of time you want to migrate to windows to web application then again you need to write code for all 1000 screens OR lets suppose tomorrow you database provider changes from sql server to oracle server then again where ever you have written sql connection in all 1000 screens in all that you need to make changes. Frankly speaking changing code in 1000 screens or re-writing code in 1000 screens is not easy it is time and money consuming job because your client or company will not pay that much or give that much time to code every time.

Difficult to understand : If a new developer joins you for the same project then it becomes very difficult to make him/her understand about the code and project (inspite of having documentation).

No proper templatization : It becomes very difficult to find any specific part of project code.

Security Threat : Our database server is not secured because database server code is directly written in the page level.

We are human beings we cannot remember all those codes which we have written in the different pages over the years (Large projects). So in order to maintain flexibility, migration facility, data security, proper maintenance and to understand project easily even after the years it is best practices to use 3 tier architecture in projects.

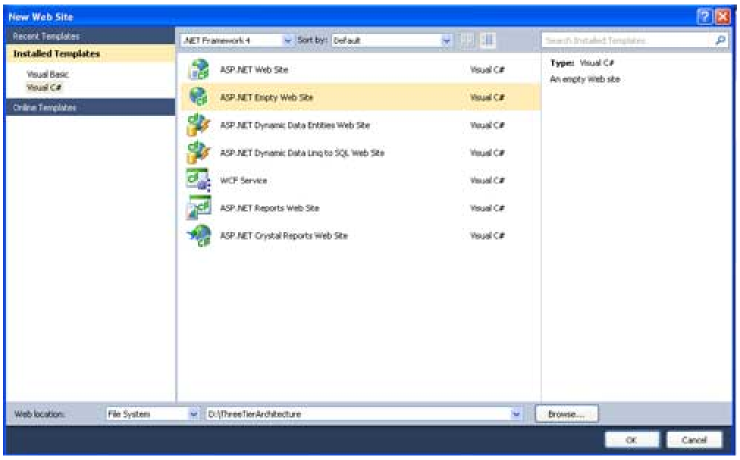
For small projects with 5 to 20 screens it is not necessary to implement three tier architecture. But for large projects it is recommended to use three tier implementation.

Step by step 3 tier architecture example

In this demonstration of the code we will use website as our presentation layer and class library project for both business layer and database layer.

Step 1 : Create Presentation Layer

We will create a website for our presentation layer.



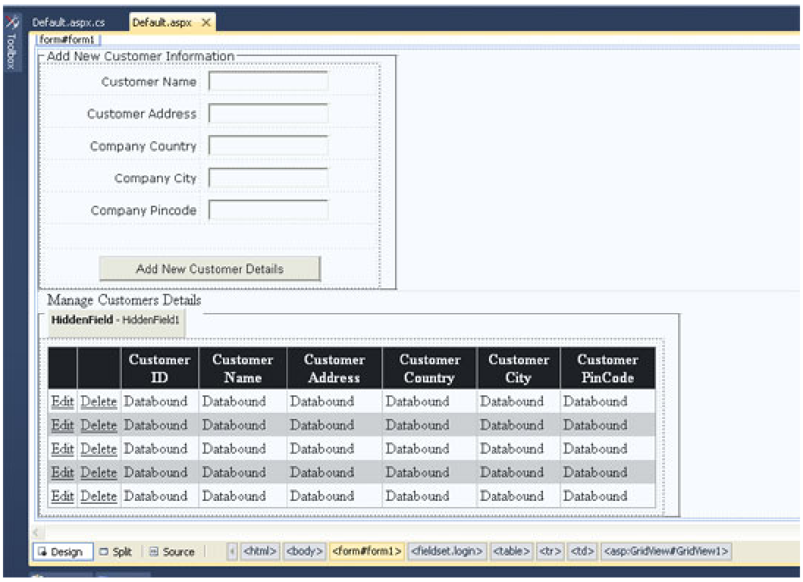
New Web Site
Recent Templates
Installed Templates
Visual Basic
Visual C#
Online Templates
Sort by:
Search Installed Templates
ASP.NET Web Site Visual C#
ASP.NET Empty Web Site Visual C#
ASP.NET Dynamic Data Entities Web Site Visual C#
ASP.NET Dynamic Data Linq to SQL Web Site Visual C#
WCF Service Visual C#
ASP.NET Reports Web Site Visual C#
ASP.NET Crystal Reports Web Site Visual C#
Browse...
OK
Cancel

.NET Framework 4
Sort by: Default
Type: Visual C#
An empty Web site
Web location: File System D:\ThreeTierArchitecture

As you see from above code that we have successfully created a website with dotnet framework 4.0 and language c#.

In our website screen we have created a customer entry screen to add a new customer to the data base means using this screen we will see how to post data over 3 tier architecture and update it in the database.

Then we have added a gridview to the screen to display all customers from the data base means using gridview screen we will see how to fetch all records from the database and display it on screen with edit and delete options.



Add New Customer Information

Customer Name
Customer Address
Company Country
Company City
Company Pincode

Manage Customers Details

		Customer ID	Customer Name	Customer Address	Customer Country	Customer City	Customer PinCode
Edit	Delete	Databound	Databound	Databound	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound	Databound	Databound	Databound

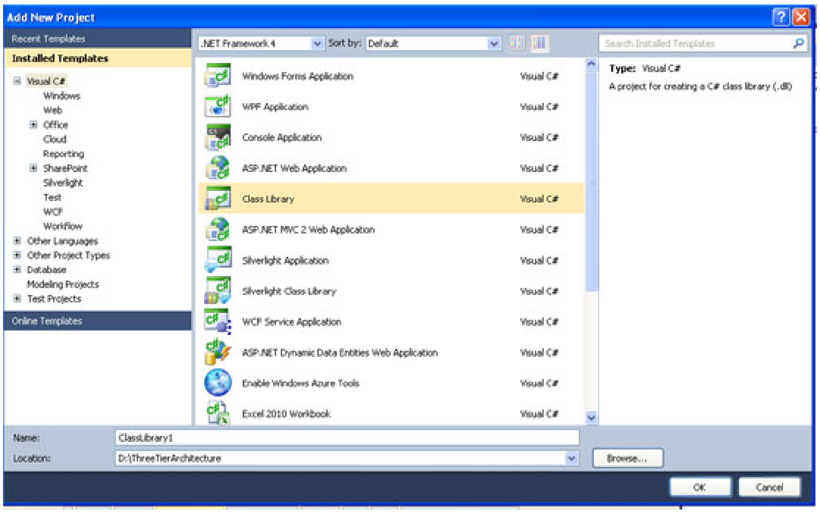
In the same gridview we provided two options "**edit**" and "**delete**" to edit and delete selected customer records. Kindly Note : If you want to know more about insert, update, delete and select in asp.net c# then click here to view my article.

"**Edit**" option provided to see how to fetch selected customer ID record from database over 3 layers. "**Delete**" option provided to see how to delete selected customer ID record from database over 3 layers.

So this is how our presentation layer looks like now we will create our database layer and start our coding from **database layer->Business Layer->Presentation Layer.**

Step 2 : Create Database Layer

In this step we will add a new project of class library to our existing project and name it as database layer.


Add New Project
Recent Templates
Installed Templates
Visual C#
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Other Languages
Other Project Types
Database
Modeling Projects
Test Projects
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
ASP.NET Web Application Visual C#
Class Library Visual C#
ASP.NET MVC 2 Web Application Visual C#
Silverlight Application Visual C#
Silverlight Class Library Visual C#
WCF Service Application Visual C#
ASP.NET Dynamic Data Entities Web Application Visual C#
Enable Windows Azure Tools Visual C#
Excel 2010 Workbook Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating a C# class library (.dll)

Name: ClassLibrary1

Location: D:\ThreeTierArchitecture

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace DatabaseLayer
{
    public class clsDataLayer
    {
    }
}
```

As you see from our above code that we have successfully created our database layer.

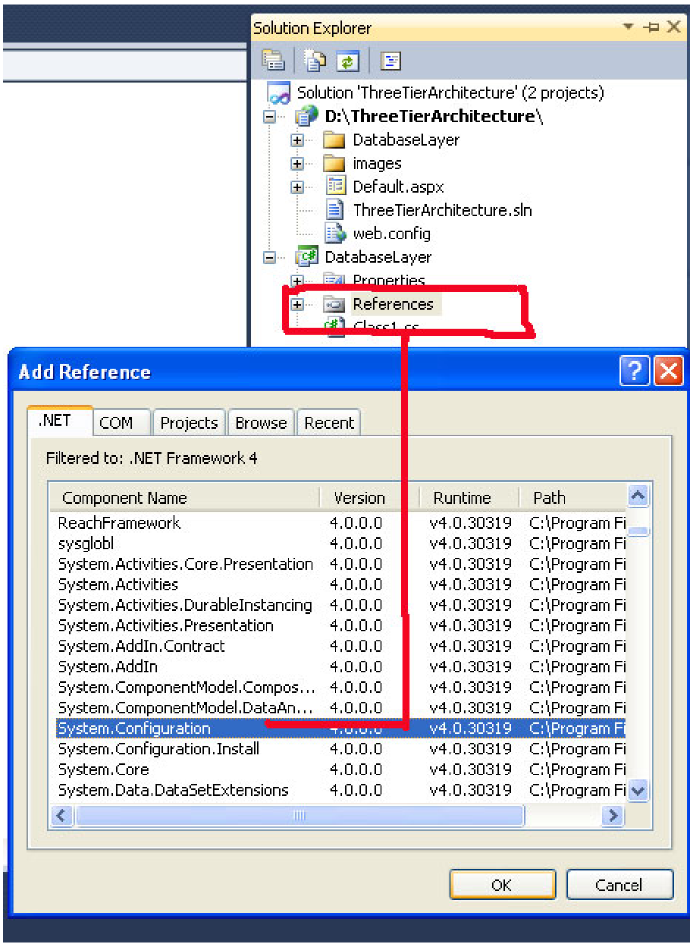
Step 3 : Create a connection to Data Provider

In our web application we will use SQL SERVER as our data provider so lets make a connection to sql server. For best practices we will write our sql connection string in the web config file. As we know that web config is the configuration file for web applications so if we write sql connection in that file we can access sql connection string from any file in the project.

```
<?xml version="1.0"?>

<configuration>

  <connectionStrings>
    <add name="Mythreetier" connectionString="Data Source=guru\sqlexpress;Initial Catalog=ProjectBilling;Integrated Security=True" providerName="System.Data.SqlClient"/>
  </connectionStrings>


</configuration>
```

So as you see from above code that we have successfully added connection string "**Mythreetier**" to web config file. Now lets call that sql connection string in database layer file.

To call connection string from web config file to database layer, manually we need to add references of **System.Configuration** to database layer references folder as shown in below image file.



Filtered to: .NET Framework 4

Component Name	Version	Runtime	Path
ReachFramework	4.0.0.0	v4.0.30319	C:\Program Fi
sysglobl	4.0.0.0	v4.0.30319	C:\Program Fi
System.Activities.Core.Presentation	4.0.0.0	v4.0.30319	C:\Program Fi
System.Activities	4.0.0.0	v4.0.30319	C:\Program Fi
System.Activities.DurableInstancing	4.0.0.0	v4.0.30319	C:\Program Fi
System.Activities.Presentation	4.0.0.0	v4.0.30319	C:\Program Fi
System.AddIn.Contract	4.0.0.0	v4.0.30319	C:\Program Fi
System.AddIn	4.0.0.0	v4.0.30319	C:\Program Fi
System.ComponentModel.Compos...	4.0.0.0	v4.0.30319	C:\Program Fi
System.ComponentModel.DataAn...	4.0.0.0	v4.0.30319	C:\Program Fi
System.Configuration	4.0.0.0	v4.0.30319	C:\Program Fi
System.Configuration.Install	4.0.0.0	v4.0.30319	C:\Program Fi
System.Core	4.0.0.0	v4.0.30319	C:\Program Fi
System.Data.DataSetExtensions	4.0.0.0	v4.0.30319	C:\Program Fi

We also need to import data references in the database layer file and finally calling connection string "**Mythreetier**" in database layer file as shown below code.

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Configuration;
using System.Data;
using System.Data.SqlClient;


namespace DatabaseLayer
{
    public class clsDataLayer
    {
    private string conn =
ConfigurationManager.ConnectionStrings["Mythreetier"].ToString();
    }
}
```

Step 4 : Create SQL Methods

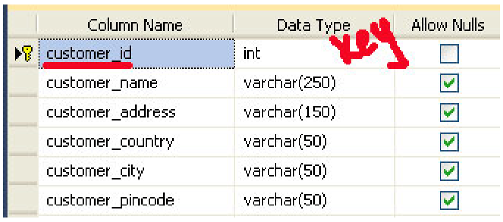
Now in this step we will create methods which will execute **Insert/Update/Delete** command using **SQLCOMMAND** and another method which execute Select command using SQLCOMMAND and **DATASET** to load all records and pass it to business layer.

```
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Configuration;
using System.Data;
using System.Data.SqlClient;


namespace DatabaseLayer
{
    public class clsDataLayer
    {
    private string conn =
ConfigurationManager.ConnectionStrings["Mythreetier"].ToString();

```

```
        public void InsertUpdateDeleteSQLString(string sqlstring)
        {
            SqlConnection objsqlconn = new SqlConnection(conn);
            objsqlconn.Open();
            SqlCommand objcmd = new SqlCommand(sqlstring, objsqlconn);
            objcmd.ExecuteNonQuery();

        }

        public object ExecuteSqlString(string sqlstring)
        {
            SqlConnection objsqlconn = new SqlConnection(conn);
            objsqlconn.Open();
            DataSet ds = new DataSet();
            SqlCommand objcmd = new SqlCommand(sqlstring, objsqlconn);
            SqlDataAdapter objAdp = new SqlDataAdapter(objcmd);
            objAdp.Fill(ds);
            return ds;
        }

    }
}
```

As you see from our above code that we have created two following methods "**InsertUpdateDeleteSQLString**()" to execute Insert/Update/Delete and "**ExecuteSqlString**()" to execute Select command.

Before we create methods for insert/update/delete/select sql queries let me just show you how our "**customer**" table structure in sql database looks like.



Column Name	Data Type	Allow Nulls
customer_id	int	□
customer_name	varchar(250)	☑
customer_address	varchar(150)	☑
customer_country	varchar(50)	☑
customer_city	varchar(50)	☑
customer_pincode	varchar(50)	☑

Now lets create methods for insert/update/delete/select sql queries as shown below.

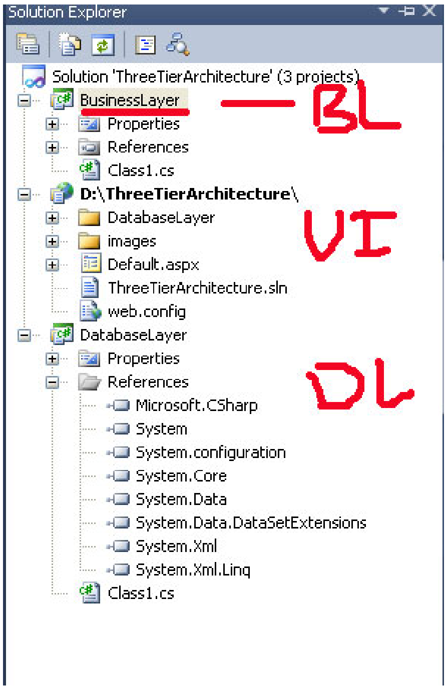
```
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.Configuration;
using System.Data;
using System.Data.SqlClient;


namespace DatabaseLayer
{
    public class clsDataLayer
    {
    private string conn =
ConfigurationManager.ConnectionStrings["Mythreetier"].ToString();

        public void InsertUpdateDeleteSQLString(string sqlstring)
        {
            SqlConnection objsqlconn = new SqlConnection(conn);
            objsqlconn.Open();
            SqlCommand objcmd = new SqlCommand(sqlstring, objsqlconn);
            objcmd.ExecuteNonQuery();

        }

        public object ExecuteSqlString(string sqlstring)
        {
            SqlConnection objsqlconn = new SqlConnection(conn);
            objsqlconn.Open();
            DataSet ds = new DataSet();
            SqlCommand objcmd = new SqlCommand(sqlstring, objsqlconn);
            SqlDataAdapter objAdp = new SqlDataAdapter(objcmd);
            objAdp.Fill(ds);
            return ds;
        }

       public void AddNewCustomerDB(string custname, string custaddr, string
custcountry, string custcity, string custincode)
        {
            DataSet ds = new DataSet();
            string sql = "INSERT into Customer
(customer_name,customer_address,customer_country,customer_city,customer_pincode)
VALUES ('" + custname + "','" + custaddr + "','" + custcountry + "','" +
custcity + "','" + custincode + "')";
            InsertUpdateDeleteSQLString(sql);
        }

        public void UpdateCustomerDB(int custid, string custname, string
custaddr, string custcountry, string custcity, string custincode)
        {
            DataSet ds = new DataSet();
            string sql = "Update Customer set customer_name='" + custname +
"',customer_address = '" + custaddr + "',customer_country= '" + custcountry
+ "',customer_city = '" + custcity + "',customer_pincode = '" + custincode +
"' Where customer_id = '" + custid + "' ";
            InsertUpdateDeleteSQLString(sql);
        }

        public void DeleteCustomerDB(int custid)
        {
            DataSet ds = new DataSet();
            string sql = "Delete From Customer Where customer_id = '" +
custid + "' ";
            InsertUpdateDeleteSQLString(sql);
        }


        public object LoadCustomerDB()
        {
            DataSet ds = new DataSet();
            string sql = "SELECT * from Customer order by customer_id";
            ds = (DataSet)ExecuteSqlString(sql);
            return ds;
        }

    }
}
```

Note : Above code is Complete Database Layer Code.

This is the complete code of database layer file as you see from above code that we have created different methods for insert/update/delete/select sql queries. Now these are the methods which will used in the business layer for customer insert, update, delete and select.

Step 5 : Create Business Layer

In this step we will add a new project of class library to our existing project and name it as business layer.



Solution Explorer

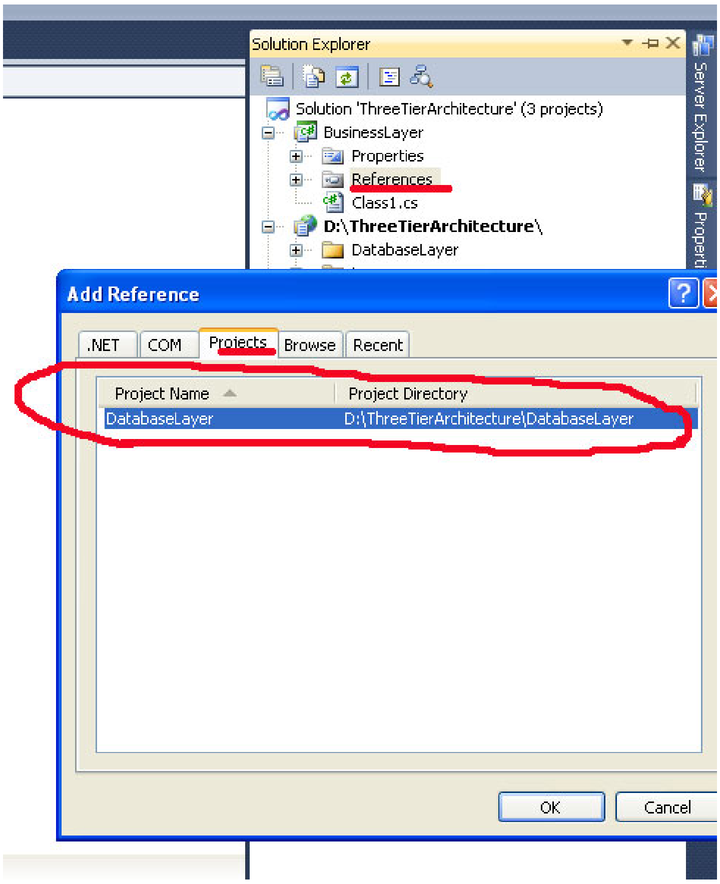
- Solution 'ThreeTierArchitecture' (3 projects)
  - BusinessLayer
    - Properties
    - References
    - Class1.cs
  - D:\ThreeTierArchitecture\
    - DatabaseLayer
    - images
    - Default.aspx
    - ThreeTierArchitecture.sln
    - web.config
  - DatabaseLayer
    - Properties
    - References
      - Microsoft.CSharp
      - System
      - System.configuration
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Xml
      - System.Xml.Linq
    - Class1.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BusinessLayer
{
    public class clsBusinessLayer
    {
    }
}
```

As you see from our above code that we have successfully created our business layer.

Step 6 : Import Datalayer Namespace in Business Layer

In this step we need to import the namespace of database layer and then create the object of class "**clsDatalayer**" as shown below code. To import the namespace manually we need to add database layer project references in the business layer references folder. as shown below.



Project Name	Project Directory
DatabaseLayer	D:\ThreeTierArchitecture\DatabaseLayer

After adding the references we need to create a database layer class object in the business layer class.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using DatabaseLayer;


namespace BusinessLayer
{
    public class clsBusinessLayer
    {
        public clsDataLayer objDataLayer = new clsDataLayer();
    }
}
```

Now in further step we will create methods in business layer for select/update/delete/insert to update from presentation layer to database layer.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using DatabaseLayer;


namespace BusinessLayer
{
    public class clsBusinessLayer
    {
        public clsDataLayer objDataLayer = new clsDataLayer();

        public void AddNewCustomer(string custname, string custaddr, string custcountry, string custcity, string custincode)
        {
            objDataLayer.AddNewCustomerDB(custname, custaddr, custcountry, custcity, custincode);
        }

        public void UpdateCustomer(int custid, string custname, string custaddr, string custcountry, string custcity, string custincode)
        {
            objDataLayer.UpdateCustomerDB(custid, custname, custaddr, custcountry, custcity, custincode);
        }

        public void DeleteCustomer(int custid)
        {
            objDataLayer.DeleteCustomerDB(custid);
        }

        public object LoadCustomer()
        {
            return objDataLayer.LoadCustomerDB();
        }
    }
}
```

Note : Above code is the complete snippet foor business layer.

So as you see from above code that we have successfully created each methods for select/insert/delete/update i.e.

For Select : LoadCustomer();
For Insert : AddNewCustomer();
For Update : UpdateCustomer();
For Delete : DeleteCustomer();

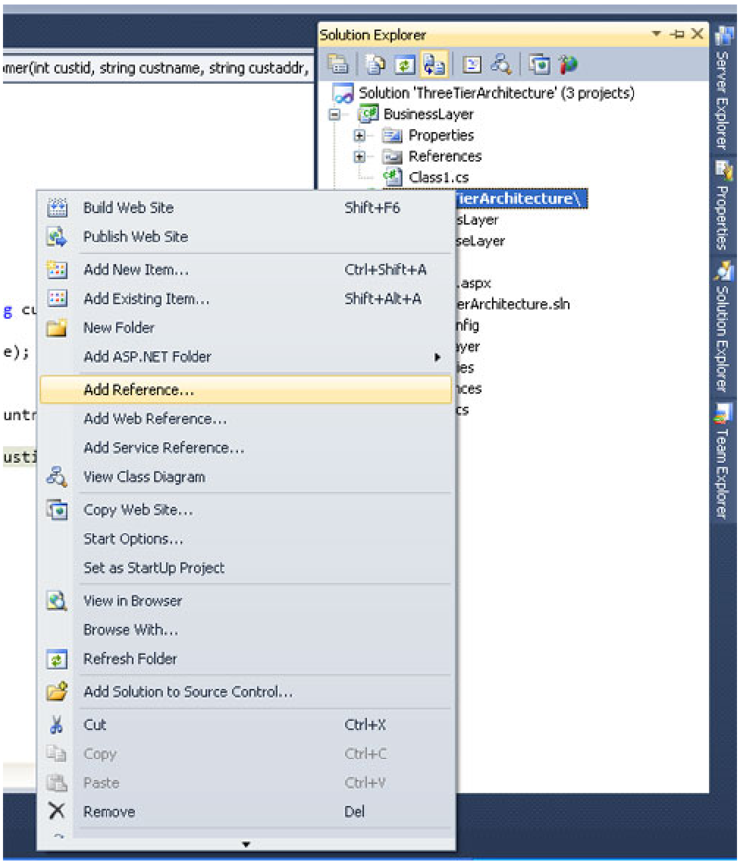
So our business layer and datalayer is now ready to take input values from presentation layer.

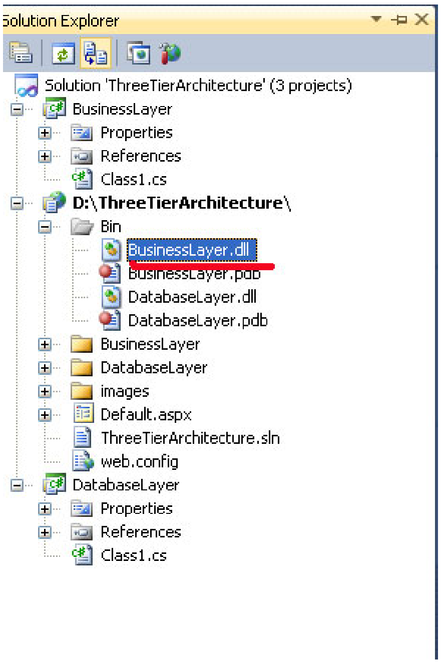
If you see closely look without touching user interface we have coded our business logic and database logic so tommorow if our user interface changes from web application to mobile application then our business logic and database logic remains the same because our business layer and database layer are not coupled with user-interface layer.

So friends as you see this is one of the main advantage of using 3 tier architecture in projects because it gives you more flexibility to handle projects over a time period if any user-interface comes or any code migration happens then we can handle them easily without changing our database layer and business layer snippet only we need to re-code our user-interface pages.

Step 7 : Passing Values to Business Layer

Finally we will now code our presentation layer. First we need to reference business layer project in the website application references folder and import the namespace of business layer.





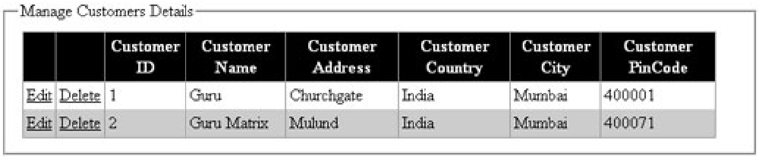
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using BusinessLayer;

public partial class _Default : System.Web.UI.Page
{
    clsBusinessLayer objLogic;

    protected void Page_Load(object sender, EventArgs e)
    {
        objLogic = new clsBusinessLayer();

    }
    protected void GridView1_SelectedIndexChanging(object sender, GridViewSelectEventArgs e)
    {

    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {

    }
    protected void btnaddcustomer_Click(object sender, EventArgs e)
    {

    }
}
```

As you see from above code that we have successfully imported the namespace in the presentation layer. Now lets do select/insert/update/delete using Business layer methods.

Select Customer

Now lets load all the customers in the Gridview using **LoadCustomer**() method.

```
clsBusinessLayer objLogic;

protected void Page_Load(object sender, EventArgs e)
{
    objLogic = new clsBusinessLayer();
  GridView1.DataSource = objLogic.LoadCustomer();
    GridView1.DataBind();

}
```



Manage Customers Details

		Customer ID	Customer Name	Customer Address	Customer Country	Customer City	Customer PinCode
Edit	Delete	1	Guru	Churchgate	India	Mumbai	400001
Edit	Delete	2	Guru Matrix	Mulund	India	Mumbai	400071

As you see from above image we have displayed all customer records in the gridview using **LoadCustomer**() method.

Insert Customer Record

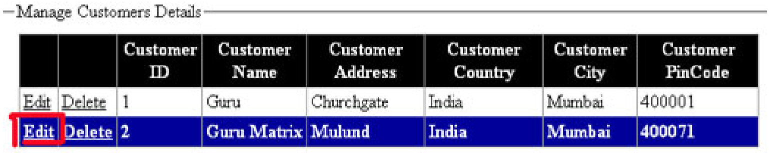
We will insert the customer record and to insert we will use **AddNewCustomer**() method in the button click event and pass appropriate values to it.

```
clsBusinessLayer objLogic;

 protected void btnaddcustomer_Click(object sender, EventArgs e)
{
    objLogic = new clsBusinessLayer();

    if (HiddenField1.Value == "")
    {
    objLogic.AddNewCustomer(txtcustname.Text, txtcustaddr.Text, txtcustcountry.Text, txtcustcity.Text, txtcustincode.Text);
    }

    Response.Redirect("Default.aspx");
}
```

Note : HiddenField is by default empty when the user clicks on edit link then we have set the HiddenField value with the customer ID but by default it is empty.

Update Customer

When the user from the screen clicks on edit link which is located on gridview then on **GridView1_SelectedIndexChanging**() event we have set all the values to text fields as shown below.



Manage Customers Details

		Customer ID	Customer Name	Customer Address	Customer Country	Customer City	Customer PinCode
Edit	Delete	1	Guru	Churchgate	India	Mumbai	400001
Edit	Delete	2	Guru Matrix	Mulund	India	Mumbai	400071

```
protected void GridView1_SelectedIndexChanging(object sender, GridViewSelectEventArgs e)
{
    int index = Convert.ToInt32(e.NewSelectedIndex);
    txtcustname.Text = GridView1.Rows[index].Cells[3].Text;
    txtcustaddr.Text = GridView1.Rows[index].Cells[4].Text;
    txtcustcountry.Text = GridView1.Rows[index].Cells[5].Text;
    txtcustcity.Text = GridView1.Rows[index].Cells[6].Text;
    txtcustincode.Text = GridView1.Rows[index].Cells[7].Text;
    HiddenField1.Value = GridView1.Rows[index].Cells[2].Text;

    btnaddcustomer.Text = "Update Customer";
}
```

If the hiddenfield contains the value and user clicks on update button then we used **UpdateCustomer**() method to update the records to the database as shown in below code.

```
clsBusinessLayer objLogic;

protected void btnaddcustomer_Click(object sender, EventArgs e)
{
    objLogic = new clsBusinessLayer();
    if (HiddenField1.Value != "")
    {

        objLogic.UpdateCustomer(Convert.ToInt16(HiddenField1.Value),
txtcustname.Text, txtcustaddr.Text, txtcustcountry.Text, txtcustcity.Text,
txtcustincode.Text);
    }

    Response.Redirect("Default.aspx");
}
```

Delete Customer

When the user clicks on delete link located on gridview then on **GridView1_RowDeleting**() click event we have called **DeleteCustomer**() method to delete a record as shown in below code.

```
clsBusinessLayer objLogic;

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs
e)
{
    objLogic = new clsBusinessLayer();

    int index = Convert.ToInt32(e.RowIndex);

    int custid = Convert.ToInt16(GridView1.Rows[index].Cells[2].Text);

    objLogic.DeleteCustomer(custid);
}
```

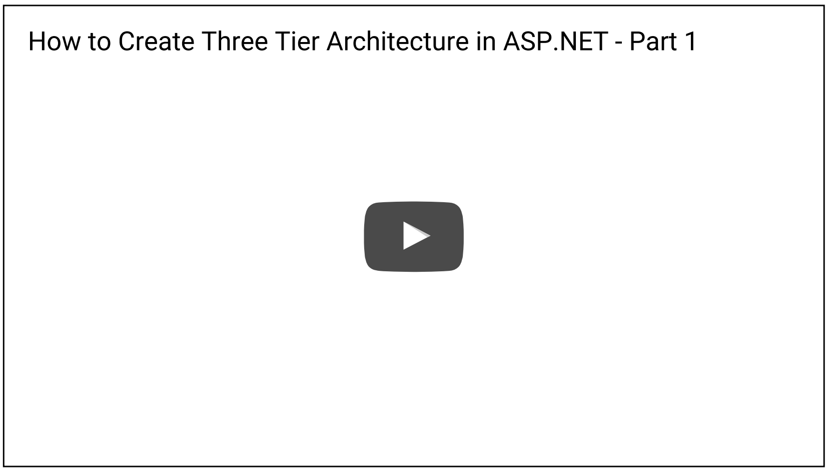
Complete Code of Presentation Layer is shown below

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using BusinessLayer;

public partial class _Default : System.Web.UI.Page
{
    clsBusinessLayer objLogic;

    protected void Page_Load(object sender, EventArgs e)
    {
        objLogic = new clsBusinessLayer();
        GridView1.DataSource = objLogic.LoadCustomer();
        GridView1.DataBind();

    }
    protected void GridView1_SelectedIndexChanging(object sender,
GridViewSelectEventArgs e)
    {
        int index = Convert.ToInt32(e.NewSelectedIndex);
        txtcustname.Text = GridView1.Rows[index].Cells[3].Text;
        txtcustaddr.Text = GridView1.Rows[index].Cells[4].Text;
        txtcustcountry.Text = GridView1.Rows[index].Cells[5].Text;
        txtcustcity.Text = GridView1.Rows[index].Cells[6].Text;
        txtcustincode.Text = GridView1.Rows[index].Cells[7].Text;
        HiddenField1.Value = GridView1.Rows[index].Cells[2].Text;

        btnaddcustomer.Text = "Update Customer";
    }
    protected void GridView1_RowDeleting(object sender,
GridViewDeleteEventArgs e)
    {
        int index = Convert.ToInt32(e.RowIndex);

        int custid = Convert.ToInt16(GridView1.Rows[index].Cells[2].Text);

        objLogic.DeleteCustomer(custid);
    }
    protected void btnaddcustomer_Click(object sender, EventArgs e)
    {

        if (HiddenField1.Value == "")
        {

            objLogic.AddNewCustomer(txtcustname.Text, txtcustaddr.Text,
txtcustcountry.Text, txtcustcity.Text, txtcustincode.Text);
        }
        else
        {
            objLogic.UpdateCustomer(Convert.ToInt16(HiddenField1.Value),
txtcustname.Text, txtcustaddr.Text, txtcustcountry.Text, txtcustcity.Text,
txtcustincode.Text);

        }
        ClearAll();
        Response.Redirect("Default.aspx");
    }

    public void ClearAll()
    {
        txtcustname.Text = "";
        txtcustaddr.Text = "";
        txtcustcountry.Text = "";
        txtcustcity.Text = "";
        txtcustincode.Text = "";
    }
}
```

So friends i hope you understood this complete tutorial on Three Tier Architecture in ASP.NET using c# language. If you have any doubts or queries kindly feel free to ask me through your comments. If any part of code is wrong or missed something kindly let me know through comments. If you like this article or if you found this article useful just share it with your friends on social media networks like Facebook, twitter and Google+.

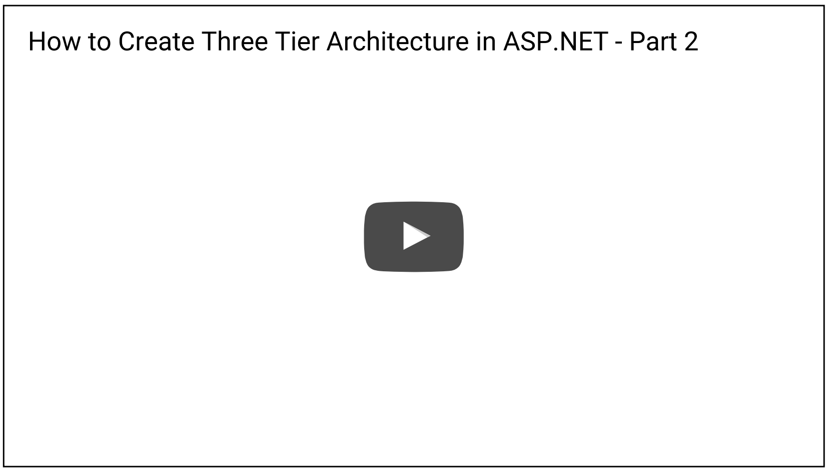
If you want to see complete online step by step tutorial on 3 tier architecture then checkout my Part 1, Part 2 and Part 3 videos on How to create 3 tier architecture in asp.net. Watch all parts of the video. (Part I, II, III).

How to Create Three Tier Architecture in ASP.NET - Part 1



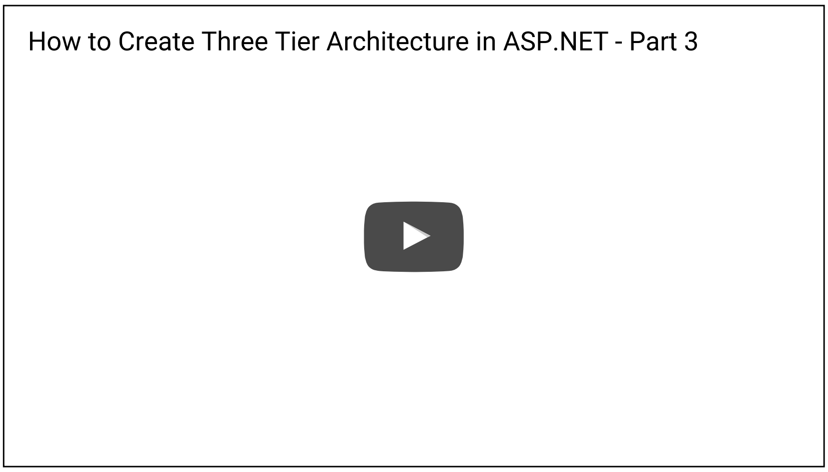
How to Create Three Tier Architecture in ASP.NET - Part 1

How to Create Three Tier Architecture in ASP.NET - Part 2



How to Create Three Tier Architecture in ASP.NET - Part 2

How to Create Three Tier Architecture in ASP.NET - Part 3



How to Create Three Tier Architecture in ASP.NET - Part 3

These above videos are also listed in my youtube channel i.e. on http://www.youtube.com/user/learnsource

Friends onlinebuff.com is on facebook, youtube channel, twitter and google plus where we regularly update our social networks with technical interview questions. So for regular updates kindly like our website on Facebook, twitter and Google + and subscribe to youtube channel.

URL to like Onlinebuff on Facebook : **https://www.facebook.com/Onlinebuff**

If you like this article or if you found this article useful just share it with your friends.

Thank you...!

:: SHARE THIS POST ::

40 Like Tweet 9 G+1 57 Share

More DOT.NET Articles To Read Next:

Sep 17 2013

Step By Step Select, Insert, Update and Delete using ASP.NET C# and ADO.NET

CLICK TO READ FULL ARTICLE

Views - **214072** | Category - **DOT.NET**

Apr

Roles of Common Language Runtime in Dot.Net Framework

18
2013

CLICK TO READ FULL ARTICLE
Views - **31451** | Category - **DOT.NET**

Gurunatha Dogi is a software engineer by profession and founder of Onlinebuff.com, Onlinebuff is a tech blog which covers topics on .NET Fundamentals, Csharp, Asp.Net, PHP, MYSQL, SQL Server, Cell Phones and lots more. Follow me @ Google+ and twitter.

Leave a Comment......

All fields marked with * are mandatory

Name *

Email *

Comment *

Security Code * **1ff9a**

Go!

Comments

Posted by **Gurunatha Dogi** on 2016-06-07

Thank you everyone...! If any query or doubts please post here...For previous post really not got time coz i was out of station...I'm back to Mumbai and will resolve all doubts..Many Thanks

Posted by **Vinit** on 2016-05-30

awesome article bro...

Posted by **Ronak Somaiya** on 2016-02-25

you deserve place in my company

Posted by **Chandrabhan Rajbhar** on 2016-02-21

it helped me a lot sir....thank u very much

Posted by **Prateek** on 2016-01-27

Sir u have not specify any validation in bl can you do it I am new to this

Posted by **Mohd Sheebu** on 2016-01-19

This article is very useful for those all who doesn't know what is 3-Tier Architecture.

Posted by **Ashutosh** on 2016-01-19

Excellent work ! nicely explained

Posted by **Kelvin** on 2015-12-09

Simple & easy to understand. i learned 3-tier in just a minute. thanks for sharing knowledge to everyone! you're God people

Posted by **Kelvin** on 2015-12-09

very nice and simple! i learned 3-tier in just a minute. thank you for sharing knowledge to everyone. God will bless you

Posted by **Rakesh Singh** on 2015-12-01

could u please give an example over interface in three tier architecture......

Posted by **Krishnaveni** on 2015-11-20

wow..great tutorial, easy to understand...good job. Thanks.

Posted by **Waghale kiran** on 2015-10-26

You have explained each and every point in a clear manner..Thank u very much

Posted by **Firoz Khan** on 2015-09-01

Hi, I need a csharp project for MTech, could you provide this. please mention how much it will cost. thanks in advance

Posted by **Chandru** on 2015-08-07

Thank You Sir Its useful

Posted by **Prabhat** on 2015-07-30

pls give me idea about mvc

Posted by **Vijay Panchasara** on 2015-05-11

Very Very Good Example I have never Seen Thanks

Posted by **Satya** on 2015-04-07

I clearly understood ..Thankyou so much sir

Posted by **Damini tripathi** on 2015-03-31

i have created this application GridView is dissable on screen... how can i solve the issue

Posted by **Priyanka** on 2014-12-29

very usefull and easy to understand.....Thank you

Posted by **Ghazi Anwar** on 2014-12-09

Dear Dogi ji I have some questions for my project which is in Asp.net. I have to install my web application where client has web server on one server, Application server on another server and database on a separate server. What we were trying to do with our existing setup is that we'll create a site on web server than point to the source code folder on app server . App server code will in turn call the data base. But client is not approving this way out. He is saying that we must have an standard 3 tier architecture according to their internal security policy. Please let me know how we can do for that. We are in a fix. If we go for state of art 3 tier architecture we have to re do the entire project. what may be the best to do it quickly . Please advise how we are going to connect these three server together or at least web server and app server. Waiting for your early response. My email id is ganwer@proind.in Thanks Ghazi Anwar

Posted by **Hardik kathiriya** on 2014-12-06

superb sir keep it up upload more this type of content so we can improve our knowledge thanks...it's very helpfull.:)

Posted by **Salim Ansari** on 2014-11-27

Hi sir, you are nice define to step by step so i am satisfy. Thanks

Posted by **CLR** on 2014-11-24

BEST EXAMPLE WITH CLARITY

Posted by **Manickam** on 2014-10-08

I Need Full Explanation with All Steps For Asp.Net 3 tier Architecture with All Screen Shots

Posted by **Gayani** on 2014-10-01

Great article sir. You have explained each and every point in a clear manner. Its easy to understand for beginners. Thank you.

Posted by **Vijetha** on 2014-09-19

Hi Guru, Need a help regarding one of the interview question asked to me. Say a table has EmpId,EmpName etc.. How do we pass more than one id(instead of executing the query or store procedure again and again) to fetch data. Its wrt ASP.net application and ADO.net data access technology. Kindly Help me out. Thanks & Regards..VJ

Posted by **S k verma** on 2014-09-16

Excellent Topic with description i like too much finally thanks a lot Dogi ji because of clear clarification. but can you provide source code for the same.

Posted by **Shaikh Noman Nasir** on 2014-09-04

Wohoo..!! Unbelievable... Simply great explanation level! You've no idea how much you helped me through posting this article.. thanks man.. God Bless You.!

Posted by **Gaurav Awasthi** on 2014-09-01

Excellent explanation For Beginners. Thanks!!

Posted by **Sharan** on 2014-08-31

i have a basis knowledge in .net frame work, i need a more clarity about making a project in 3 tier arch.. and how to set up the file and how to launch in server

Posted by **NAYAB RASOOL SHAIK** on 2014-08-15

super bro excellent it is helped me lot thank u bro keep posting like this

Posted by **Kumar** on 2014-08-07

how to hide popup window in asp.net

Posted by **Arvind kumar yadav** on 2014-08-06

This is very good topic for new who want to learn Three tier code Thank you so much.

Posted by **Amit** on 2014-08-05

Very Nice......Thank You.....!!!!

Posted by **Kanudawala nilesh** on 2014-08-05

very very more Excellent...........!

Posted by **Shrikant Yadav** on 2014-08-03

This article proved really usefull which has made my all doubts clear regarding 3-tier Architecture at a glance. Thanks

Posted by **Avinash** on 2014-07-23

Great Post :) can u please implement the same using properties.

Posted by **Rajesh** on 2014-07-22

Nice Bro,... I will share the links with myfriends who needs to know about 3 tier architecture. Can I???

Posted by **Arun** on 2014-07-18

excellent work sir

Posted by **Sobha** on 2014-07-15

Excellent

Posted by **Bhuva Bhavesh** on 2014-05-30

very nice thank you..

Posted by **Sufyan Ali** on 2014-05-23

extra ordnry explain bro..........

Home