# **ADO.NET**

What is ADO.Net?

ADO.NET is .Net data access Technology.

Why ADO.Net?

To Communicate Database

What is Database

Representing the collection of data in tables format is called as Database.

What is Table?

Representing the data in rows and columns format is called as Table.

Why database or purpose of database?

To store the user information for future access.

An Application can be divided into two parts

1. FrontEnd part
2. Backend part

**FrontEnd Part:**

This part of application will accept input from the user as well as it will display output to the user.

End user is directly interacting the frontend part.

FrontEnd part of the application can be developed by using frontend technologies called .Net, Java,php…..

**BackEnd Part:**

This part of the application will store the user information.

To develop backend part of the application we have to use backend technologies like SQL server,Oracle, MySQL, MSAccess……

These backend technologies are called as RDBMS(Relational database Management Systems).

To allow the communication between frontend and backend we do required a middle man called Data Access Object.

**Data Access Object**

Data Access object allow the communication between frontend and backend.

.Net data access object is ADO.NET.

Java Data access object is JDBC.

**Diagram to represent frontend &**

**Backend**

Front End

part

Data Access Object

Back end part

.Net

ADO.NET

SQL SERVER

Java

JDBC

Oracle

ADO.Net is .Net data access Technology which was introduced .Net framework 1.0. Whenever a .Net application wants to communicate database it has to take the help of ADO.Net.

ADO.Net is a mediator between .Net application and database.

**DataAccess Architecture**

The Style how the frontend part of the application will communicate backend part of the application will be representing by data access architecture.

These Data access Architectures are 2 types

1. Connected Oriented Architecture.
2. Disconnected Oriented Architecture.

## Connected Oriented Architecture

In Connected Oriented Architecture client application will perform database operations on central database directly.

Before performing database operation connection has to open explicitly to central database as well as after completion of the database operation connection has to close explicitly. In this architecture programmer has to write the code to open the connection as well as to close the connection.

**Diagram For Connected Oriented Architecture**

Client Application central databases

Connection is open

S

s

elect \* from Emp

G

g

etting emp table

Closing connections

EmpNo EmpName sal

1

Ravi 1000

2

Raj

2000

Display

EmpNo EmpName sal

1

Ravi 1000

2

Raj 2000

**Advantages**

Accessing data will be faster.

**DisAdvantages**

It increase network Traffic Burden on Database server.

Explicitly programmer has to write the code to open the connection as well as to close the connection.

To overcome this above connected oriented architecture we have to go for disconnected oriented architecture.s

**Disconnected Oriented Architecture**

In this architecture client application will not perform all the database operations on central database instead of that it will depend on local database.

 Central database will be part of remote machine but local database will be part of client machine.

Whenever client application required some data it will fetch from the central database, will be fill into local database then only we can bind the data from local database to client application.

If we want to perform any modifications like update, insert, delete, select. In this architecture we will perform on local database which will reflect to central database directs(internally). In this architecture programmer does not require to open the connection explicitly as well as no need to write any code to close connection. This will happen internally automatically.

**Diagram for disconnected Oriented Architecture**

Central databases

Select \* from Emp

Inserting info into employee

local database

getting Emp Table

EmpNo EmpName sal Add/Save

1

Ravi 1000

Add

2

Raj 2000 Add

3

Sathya 3000 Save

EMP

EmpNo EmpName sal

1

Ravi 1000

2

Raj 2000

EmpNo EmpName sal

1

Ravi 1000

2

Raj 2000

**Advantages**

It reduces network traffic.

Reduce the burden on database server which improves the performance of the application.

No need to write the code to open (or) to close the connection.

**Drawbacks**

Accessing the data will be slow.

In this Architecture local database object is creating within the client machine which consumes client machine memory which affects the performance of client application.

**Evaluation of ADO.NET:**

Before .NET, under Microsoft family we have 3 popular technologies they are

1. VB
2. VC++
3. ASP

For these Technologies Microsoft has introduced Data Access objects like below

1. DAO
2. RDO
3. ADO

All the above will support only connected oriented architecture.

Due to that reason while introducing .Net by Microsoft, they have introduced a new Data Access Object called ADO.Net which will support both architectures i.e,

1. Connected Oriented Architecture
2. Disconnected Oriented Architecture.

Finally ADO.NET will depend on two components

1. Data Provider
2. Dataset

**DataProvider**

Data Provideris responsible for two things.

1. Providing the connection to central database.
2. Selecting SQL Commands(select, Insert, Update…..)
3. Fetching the SQL commands result supplying to client application on filling into local database.
4. To communicate various databases Microsoft is providing various data provider as base class libraries like below

Using System.Data.SqlClient;

Using System.Data.Oledb;

Using System.data.Odbc;

In every data provider base class library to full fill the above

Microsoft has provided 4 predefined classes like below

1. SqlConnection
2. SqlCommand
3. SqlReader
4. SqlDataAdapter

**Structure of System.Data.SqlClient base class library**

Namespace SystemDataClient

{

Class SqlConnection

{

}

Class SqlCommand

{

}

Class SqlReader

{

}

Class SqlAdapter

{

}

}

Whenever we want to use data provider we have to create 4 objects. They are

1. Connection object
2. Command Object
3. DataReader Object
4. DataAdapter Object

**Connection Object:**

How to create connection object?

SqlConnection conn=new SqlConnection();

How to open the connection to central database?

Conn.Open();

**Open():**

It is a predefined member method of sqlconnection class.

This method will open the connection to central database.

How to close the connection?

Conn.close();

**Command Object:**

How to create command object?

SqlCommand cmd=new SqlCommand();

Using command object we can execute the sqlcommand with in the Sqlserver database.

**DataReader**

How to create DataReader object

SqlDataReader dr=cmd.ExecuteReader();

Using datareader object we can fetch the SqlCommand object execution results and we can supply to client application.

**DataAdapter Object**

How to create DataAdapter object.

SqlDataAdapter da= new SqlDataAdapter();

Using DataAdapter we can fetch the SqlCommand object execution results & forwarding towards to client Application then filling into local database.

DataAdapter will be acting as a mediator between central database &local db like below

Local db

Data Adapter

Central db

DataSet:

Dataset is nothing but local database which we will use only in disconnected oriented architecture to store the fetched data from central database. How to create local database?

DataSet ds=new DataSet();

DataSet object will contain collection of tables.

Dataset will create internally in xml fromat.

DataSet: It is predefined class which is part of the system Data BCL and it is not part of any Data Provider.

What are the components we will use in connected oriented architecture.

1. Data Provider Connection object

Command object

DataReader object

What are the components we will use in Disconnected oriented architecture.

1. Data Provider Connection object

Command object

Data Adapter object

1. DataSet

Step by step process to communicate sql server DB by using disconnected oriented architecture.

Step1://Importing Data Provider

Using System.Data.SqlClient;

Using System.Data;

Step2: // Declaring connection string

String cs=”server=.; database=MyDataBse; uid=sa; pwd=abc; ” Connection string will contain 4 fields.

1. Server

We have to initialize the name of the machine which contains SqlServer.

1. DataBase

How we have to initialize the database name which we are going to communicate.

Uid: here we have to initialize the user id of the sql server.

Pwd: here we have to initialize the password of the sql sever.

Step3: // creating connection object

SqlConnection conn=new SqlConnection(cs) ConnectionString

Step4: //Defining command object by initializeing the SalCommand String

SqlCommand cmd=new SqlCommand(“select \*from Student",conn)

Step5: //creating local database

DataSet ds = new DataSet();

Step6://Creating dataAdapter object by initializing command object

SqlDataAdapter da = new SqlDataAdapter(cmd); Step7: //Filling the fetched data into local database da.Fill(ds,”Empnew”);

Step8: A/C to requirement we will perform the operation on local database.

1. Inserting the record from user interface to local data base.
2. Modifying the record in local database from user interface.
3. Removing the record in local database from user interface.
4. If the requirement is only displaying binding the data from local data base to user interface.

Step by step process of sql server data base using connected oriented.

Step1: // importing data provider

Using System.Data.SqlClient;

Step2: // Declaring connection object

String cs=”server=.; database=MyDataBse; uid=sa; pwd=abc; ”

Step3: // creating connection object

SqlConnection conn=new SqlConnection(cs) ConnectionString

Step4: // opening connection

Conn.Open();

Step5: // creating command object

SqlCommand = new SqlCommand(“select \*from emp”,conn);

Step6:// creating DataReader object

SqlDataReader dr=cmd.ExecuteReader();

Step7:// Binding data from DataReader object to client application user interface controls.

Step8: // closing the connection

Conn.Close();

ADO.Net Architecture

Ado.Net Architecture will represent components which we will use to communicate the database.

**Diagram for Ado.Net Architecture**

Data Provider DataSet

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | • | Connection Object | | • | Command Object | | • | Data Provider Object | | • | Data Adapter Object | | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | DataBase Collection   |  |  |  |  |  | | --- | --- | --- | --- | --- | | DataBase   |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | Data Row Collection | | Data Column Collection | | Data Constraints Collection | | | | |
| Central DataBase |
|  |
| |  | | --- | | XML | |

DataSet:

Dataset will store the fetched data from central data base in table fromat which are called as data tables and between tables we will maintain the relations which are called as data realtions.

Finally we can say dataset is a collection of data rows data column and data constraints. Data table is a collection of data rows data columns and data constraints.

ADO.Net Programming in Asp.Net:

ADO will support two access models

1. Wizard Model
2. Programming Model

Wizard Model:

To communicate database by using wizard model programmer need not to write the code. This wizard model we will use to test the data base connections.

Programming Model:

Using programming model if we want to communicate database we have to write the program. Programming model we can implement by using 2 architectures.

1. Disconnected Oriented Architecture
2. Connected Oriented Architecture

DataBind Controls:

Using databound controls we can display data in rows and column format.

When we will go for databound control?

Whenever we want to work with collection of data we will go for databound control like GridView, ListView, Repeater, DataList, FromView, DetailsView and so on…

GridView:-

Gridview is one of the popular data bound control in Asp.net because which will support various features.

Using gridview control we can display data in rows and columns fromat as well as using gridview we can perform following operations on any database like Sql, Oracle, MySql, Xml.

Example to communicate SQL Server DataBase by using wizard model in step by step process

WebForm1.aspx

EmpNo

Ename

Salary

1

2

Ravi

Raju

1000

2000

SQL Server

Emp

EmpNo

Ename

Salary

1

2

Ravi

Raju

1000

2000

MyDataBase

Create Database:

Create database mydatabase

Create a Table

Create table Emp(EmpNo int primary key, EmpName varchar(20),Salary money)

How to enter into database

Use mydatabase

Insert into emp values(1,’ravi’,1000)

Insert into emp values(2,’raju’,2000)

How to retrieve and display Emp table

Select \*from Emp

Example to communicate Sql Server by using disconnected oriented architecture.

Step1: design of fornt end and back end

Same as above using System; using System.Data; using System.Data.SqlClient;

namespace DisConnected

{

publicpartialclassWebForm1 : System.Web.UI.Page

{

protectedvoid Page\_Load(object sender, EventArgs e)

{

//connection string

String cs="Server=AMMA\\VIJAY;DataBase=mydatabase;uid=sa;pwd=0037;

//creating connction object

SqlConnection conn = newSqlConnection(cs);

//create command object

SqlCommand cmd= newSqlCommand("select \*from Emp",conn);

//creating local database

SqlDataAdapter da= newSqlDataAdapter(cmd);

DataSet ds = newDataSet() //filling into local database da.Fill(ds,”empnew”);

//binding emp table from local data base to gridview gridemp.DataSource=ds.Tables\*“empnew”+; gridemp.DataBind();

}

}

}

Example to communicate Sql server database by using conncted oriented architecture

Step1: design of front end and back end

WebForm1.aspx

EmpNo

Ename

Salary

1

2

Ravi

Raju

1000

2000

SQL Server

Emp

EmpNo

Ename

Salary

1

2

Ravi

Raju

1000

2000

MyDataBase

Display

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

namespace Connected

{

publicpartialclassWebForm1 : System.Web.UI.Page

{

protectedvoid Page\_Load(object sender, EventArgs e)

{

}

protectedvoid btndisplay\_Click(object sender, EventArgs e)

{

String cs="Server=AMMA\\VIJAY;DataBase=mydatabase;uid=sa;pwd=0037; SqlConnection conn = newSqlConnection(cs);

SqlCommand cmd= newSqlCommand("select \*from Emp",conn);

SqlDataReader dr=cmd.ExecuteReader();

}

}

}

Example for to communicate SQL server Database by using Disconnected Oriented Architecture

GRID VIEW CONTROL:

1. Gridview is one of the data bound control, using this control we display the data in rows and columns format.

1. Whenever we want to display the data in rows and columns format within the page then we can use Gridview control.

Step 1: Webform1.aspx SqlServer

Gri

Step 2 : Webform1.aspx.cs

//Step 1)

using

System.Data;

using

System.Data.SqlClient;

GridEmp

MydataBase

EmpNo

EmpName

sal

1

Ravi

1000

2

Vinod

2000

3

Venkat

3000

Emp

namespace DisconnectedExpmple

{

PublicpartialclassWebForm1 : System.Web.UI.Page

{

Protectedvoid Page\_Load(object sender, EventArgs e)

{

// Step2 string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

// Step3SqlConnection conn = newSqlConnection(cs) conn.Open();

// Step4SqlCommand cmd = newSqlCommand("select \* from emp",conn);

//Step5SqlDataAdapter da = newSqlDataAdapter(cmd);

DataSet ds = newDataSet(); da.Fill(ds, "empnew"); conn.Close(); // Step7 // Binding

Gridemp.DataSource = ds.Tables["empnew"];

Gridemp.DataBind();

}

}

}

 Example to communicate the SQLServer database with the help of windows Authentication and using connected Oriented Architecture.

Step 1:Webform1.cs[design]

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| EmpNo | EmpName | Salary |  |
| 1 | Ravi | 1000 |
| 2 | Vinod | 2000 |
| 3 | Venkat | 3000 |
| |  | | --- | | Dispaly |   gridEmp  btndisplay | |  |

Setp2: webform1.aspx.cs

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

namespace WebApplication1

{

PublicpartialclassWebForm1 : System.Web.UI.Page

{

Protectedvoid Page\_Load(object sender, EventArgs e)

{

// Setp1string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

// Setp2SqlConnection conn = newSqlConnection(cs);

conn.Open();

// Setp3 SqlCommand cmd = newSqlCommand("select \* from emp",conn);

// Setp4 SqlDataReader dr = cmd.ExecuteReader();

// Setp5 Binding From data Reader to gridview

Gridemp.DataSource = dr; Gridemp.DataBind();

conn.Close();

}

}

}

 Example to insert a record into SQL server database

Step 1:Webform1.aspx

Insert(panel1)

EmpNo

txtempno

EmpName

txtempname

Salary

txtsal

btninsert

Insert

lblinsert

EmpNo EmpName Sal

1

Ravi

1000

2

Vinod

2000

3

Venkat

3000

4

Sathya

4000

Protectedvoid btninsert\_Click(object sender, EventArgs e)

{

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = newSqlConnection(cs); conn.Open();

int eno = Convert.ToInt32(txtempno.Text); string ename = txtempname.Text; int esal = Convert.ToInt32(txtsal.Text);

SqlCommand cmd = newSqlCommand("insert into emp values(@eno,@ename,@esal)", conn); cmd.Parameters.AddWithValue("@eno",eno); cmd.Parameters.AddWithValue("@ename",ename); cmd.Parameters.AddWithValue("@esal",esal); int i= cmd.ExecuteNonQuery();

conn.Close(); if (i == 1)

{

lblinsert.Text = "Record inserted";

}

else

{

lblinsert.Text = "Record not inserted";

}

txtempno.Text = ""; txtempname.Text = ""; txtsal.Text = "";

txtempno.Focus();

}

 Example to delete a employee record within the SQLserver database Note :Constraints will maintain the consistency in the table.

Step 1:Webform1.aspx

Delete(panel)

Select EmpNo

dropeno

btndelete

DELETE

lbldelete

This requirement we can implement in two tasks

1. Binding the empno column of emp table to dropdownlist control 2. Delete selected employee record from emp table of central database.

Step2 :Webform1.aspx.cs

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

namespace DeletingExample

{

PublicpartialclassWebForm1 : System.Web.UI.Page

{

Protectedvoid Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = newSqlConnection(cs);

conn.Open();

SqlCommand cmd = newSqlCommand("select \* from emp", conn);

SqlDataAdapter da = newSqlDataAdapter(cmd);

DataSet ds = newDataSet(); da.Fill(ds, "empnew"); conn.Close();

//binding empno column of emp table from dataset to dropdown list Dropeno.DataSource = ds.Tables["empnew"];

Dropeno.DataTextField = "empno";

Dropeno.DataBind();

}

}

Protectedvoid btndelete\_Click(object sender, EventArgs e)

{

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = newSqlConnection(cs); conn.Open();

int eno = Convert.ToInt32(Dropeno.SelectedItem);

SqlCommand cmd = newSqlCommand("delete from emp where eno=@eno", conn); cmd.Parameters.AddWithValue("@eno", eno);

int i = cmd.ExecuteNonQuery(); conn.Close(); if (i == 1)

{

lblinsert.Text = "Record deleted";

}

else

{

lblinsert.Text = "Record not deleted";

}

txtempno.Text = "";

txtempno.Focus();

}

Note : In the above emp dropdownlist control is displaying deleted employee numbers bit it has to display the last employee numbers of the emp table like below

PublicpartialclassWebForm1 : System.Web.UI.Page

{

Publicvoid Bind()

{

//write the code from above pogram load event

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = newSqlConnection(cs);

conn.Open();

SqlCommand cmd = newSqlCommand("select \* from emp", conn);

SqlDataAdapter da = newSqlDataAdapter(cmd);

DataSet ds = newDataSet(); da.Fill(ds, "empnew"); conn.Close();

Dropeno.DataSource = ds.Tables["empnew"];

Dropeno.DataTextField = "empno";

Dropeno.DataBind();

}

Protectedvoid Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

{

Bind();

}

}

}

Protectedvoid btndelete\_Click(object sender, EventArgs e)

{

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = newSqlConnection(cs); conn.Open();

int eno = Convert.ToInt32(Dropeno.SelectedItem);

SqlCommand cmd = newSqlCommand("delete from emp where eno=@eno", conn); cmd.Parameters.AddWithValue("@eno", eno);

int i = cmd.ExecuteNonQuery(); conn.Close(); if (i == 1)

{

lblinsert.Text = "Record deleted";

}

else

{

lblinsert.Text = "Record not deleted";

}

txtempno.Text = "";

txtempno.Focus();

}

 Exapmle to updating the employee salary

Update emp set sal=5000 where empno=3

Step1:Webform1.aspx

s

Update

Select EmpNo

dropeno

Enter New Salary

txtnewsal

Btnupdate

UPDATE

The above example we can implementing in two tasks

1. Binding the empno column of emp table to dropdown list control
2. Update the selected employee salary within the emptable

Step 2: Webform1.aspx.cs

Public partial class WebForm1: System.Web.UI.Page

{

Public void Bind()

{

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = new SqlConnection(cs);

conn.Open();

SqlCommand cmd = new SqlCommand("select \* from emp", conn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet(); da.Fill(ds, "empnew"); conn.Close();

Dropeno.DataSource = ds.Tables["empnew"];

Dropeno.DataTextField = "empno";

Dropeno.DataBind();

}

Protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

{

Bind();

}

}

}

Protected void btnupdate\_Click(object sender, EventArgs e)

{

string cs = "server=Sathya; database=Employee; uid=sa; pwd=abc";

SqlConnection conn = newSqlConnection(cs); conn.Open();

int eno = Convert.ToInt32(Dropempno.SelectedItem);

int esal = Convert.ToInt32(txtnewsal.Text);

SqlCommand cmd = newSqlCommand("update emp set esal=@esal where eno=@eno", conn); cmd.Parameters.AddWithValue("@eno", eno); cmd.Parameters.AddWithValue("@esal", esal); int i = cmd.ExecuteNonQuery(); conn.Close(); if (i == 1)

{

lblinsert.Text = "Record updated";

}

else

{

lblinsert.Text = "Record not updated";

}

txtnewsal.Text = "";

 Example to create a database within Sqlserver from .Net application in runtime. Step 1 : Webform1.aspx

|  |  |
| --- | --- |
| Create Database  btndatabase       |  | | --- | | lbldatabase | |

Public partial classWebForm1 : System.Web.UI.Page

{

Protectedvoid btndatabase\_Click(object sender, EventArgs e)

{

string cs = "server=Sathya; uid=sa; pwd=abc"; SqlConnection conn = newSqlConnection(cs); conn.Open(); try