How do we connect FE with BE

FE > Console Application

BE > Sql Server

Follow some steps

1. Add the required namespace/library
2. Create object of SqlConnection Class(We pass connectinString here, we shud put connectionstring in app.config file)
3. Create object of SqlCommand(Here we have 2 parameters, 1st is the query, 2nd is connection object)
4. Open Connection
5. Execute Method (ExecuteReader (for Select Query), ExecuteNonQuery(DML operations) , ExecuteSaclar(For a Single Value)
6. Close Connection
7. Dispose off Command & Connection object

2. Ways 1. )Use Dispose Method 2) use using block

Till now, we have used Queries

Select \* from employee

Insert into employee values(\_)

**We can either write Queries OR we can use Stored Procedures**

**Why do we write functions**

1. **Reusablity**
2. **Modularity**

**Queries are always compiled**

**We can not call them multiple times**

**Even if we run multiple queries they will be sent to sql server separately.**

**Int add(int x, int y)**

**{return x+y;**

**}**

**Begin > {**

**End > }**

use practicedb

create procedure GetEmployees

AS

Begin

select \* from EMployee

End

-- Call it

CALL PROCEDURE

exec GetEmployees

use practicedb

create procedure GetEmployees

AS

Begin

select \* from EMployee

End

-- Call it

exec GetEmployees

create procedure InsertEmployee(@id int, @name varchar(20),

@address varchar(20), @salary int)

AS

Begin

insert into Employee values(@id, @name,@address,@salary)

End

create procedure UpdateEmployee(@id int,

@address varchar(20), @salary int)

AS

Begin

Update Employee set address =@address, salary = @salary where id=@id

End

create procedure DeleteEmployee (@id int)

AS

Begin

Delete Employee where id=@id

End

// CALLING SP THRU ADO.NET

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

using System.Configuration;

namespace AdoNetDemo

{

enum choice { Insert = 1, Update, Delete, GetRecords };

class SPDemo

{

static void Main()

{

int Choice;

string ch = "y";

while (ch == "y")

{

try

{

MainMenu();

Console.WriteLine("Enter Your Choice");

Choice = Byte.Parse(Console.ReadLine());

switch (Choice)

{

case (int)choice.Insert:

{

InsertRecord();

break;

}

case (int)choice.Update:

{

UpdateRecord();

break;

}

case (int)choice.Delete:

{

DeleteRecord();

break;

}

case (int)choice.GetRecords:

{

GetRecords();

break;

}

default:

{

Console.WriteLine("Invalid Choice");

break;

}

}

Console.WriteLine("Do you want to repeat the process");

ch = Console.ReadLine();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

}

static void MainMenu()

{

Console.WriteLine("1. Insert Record");

Console.WriteLine("2. Update Record");

Console.WriteLine("3. Delete Record");

Console.WriteLine("4. Display All Records");

}

static string GetConnectionString()

{

string connectionString = ConfigurationManager.AppSettings

["MyConnection"].ToString();

return connectionString;

}

static SqlConnection GetConnnection()

{

SqlConnection connection = new SqlConnection(GetConnectionString());

// connection = new SqlConnection("data source=LAPTOP-53S2KQS8;" +

// "initial catalog=PracticeDb1;integrated security=true");

return connection;

}

static void InsertRecord()

{

using (SqlConnection connection = GetConnnection())

{

Console.WriteLine("Enter ID");

int id = Byte.Parse(Console.ReadLine());

Console.WriteLine("Enter Name");

string name = Console.ReadLine();

Console.WriteLine("Enter Address");

string address = Console.ReadLine();

Console.WriteLine("Enter Salary");

int salary = int.Parse(Console.ReadLine());

**using (SqlCommand command = new**

**SqlCommand("InsertEmployee", connection))**

**{**

**command.CommandType = CommandType.StoredProcedure;**

command.Parameters.AddWithValue("@id", id);

command.Parameters.AddWithValue("@name", name);

command.Parameters.AddWithValue("@address", address);

command.Parameters.AddWithValue("@salary", salary);

connection.Open();

int count = command.ExecuteNonQuery();

Console.WriteLine("No of Records inserted are " + count);

connection.Close();

}

}

}

static void UpdateRecord()

{

using (SqlConnection connection = GetConnnection())

{

Console.WriteLine("Enter ID whose Record you want to modisy");

int id = Byte.Parse(Console.ReadLine());

Console.WriteLine("Enter New Address");

string address = Console.ReadLine();

Console.WriteLine("Enter Revised Salary");

int salary = int.Parse(Console.ReadLine());

**using (SqlCommand command = new SqlCommand("UpdateEmployee", connection))**

**{**

**command.CommandType = CommandType.StoredProcedure;**

command.Parameters.AddWithValue("@id", id);

command.Parameters.AddWithValue("@address", address);

command.Parameters.AddWithValue("@salary", salary);

connection.Open();

int count = command.ExecuteNonQuery();

Console.WriteLine("No of Records updated are " + count);

connection.Close();

}

}

}

static void DeleteRecord()

{

using (SqlConnection connection = GetConnnection())

{

Console.WriteLine("Enter ID whose Record you want to delete");

int id = Byte.Parse(Console.ReadLine());

**using (SqlCommand command = new SqlCommand("DeleteEmployee", connection))**

**{**

**command.CommandType = CommandType.StoredProcedure;**

command.Parameters.AddWithValue("@id", id);

connection.Open();

int count = command.ExecuteNonQuery();

Console.WriteLine("No of Records deleted are " + count);

connection.Close();

}

}

}

static void GetRecords()

{

using (SqlConnection connection = GetConnnection())

{

**using (SqlCommand command = new SqlCommand**

**("GetEmployees",**

**connection))**

**{**

**command.CommandType = CommandType.StoredProcedure;**

connection.Open();

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

while (reader.Read())

{

Console.WriteLine(reader["id"].ToString() + " " + reader["name"]);

}

}

reader.Close();

connection.Close();

}

}

}

}

}