Steps for Code First Approach

Code First Approach

1. Create Class

public class Department

{

public int Id {get;set;}

public string Name {get;set;}

public string ManagerName {get;set;}

public int EmployeesCount {get;set;}

}

2. Create class which inherits from DbContext

public class DepartmentDbContext : DbContext

{

public DbSet<Department> Departments { get;set;}

}

3. Add one Entry in app.config file

<ConnectionStrings>

<add name="DepartmentDbContext" connectionString=""/>

</ConnectionStrings>

--------------------------------------------------------------------------

**DEPARTMENT.CS FILE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeFirstApproachDemo

{

public class Department

{

public int Id { get; set; }

public string Name { get; set; }

public string ManagerName { get; set; }

public int EmployeesCount { get; set; }

}

}

**DeptContext.cs**

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeFirstApproachDemo

{

class DeptContext :DbContext

{

public DbSet<Department> Departments { get; set; }

}

}

**App.config**

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<configSections>

<!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->

<section name="entityFramework"

type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"

requirePermission="false"/>

</configSections>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2"/>

</startup>

<entityFramework>

<providers>

<provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer"/>

</providers>

</entityFramework>

<connectionStrings>

**<add name="DeptContext" connectionString="data source=LAPTOP-53S2KQS8;initial catalog=practicedb;integrated security=true" providerName="System.Data.SqlClient"/>**

**</connectionStrings>**

</configuration>

---------------------------------------------------------------------------------------

**PROGRAM.CS FILE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodeFirstApproachDemo

{

class Program

{

static DeptContext db = new DeptContext();

static void Main(string[] args)

{

GetDepartments();

Console.WriteLine("Enter ID");

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

Console.WriteLine("Enter Dept Name");

string name = Console.ReadLine();

Console.WriteLine("ENter Manager Name");

string manager = Console.ReadLine();

Console.WriteLine("Enter Employees Count");

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

InsertDepartment(id, name, manager, count);

Console.WriteLine("Enter ID taht Department to edit");

int edit\_id = int.Parse(Console.ReadLine());

Console.WriteLine("Enter Updated Dept Name");

string edit\_name = Console.ReadLine();

Console.WriteLine("ENter Updated Manager Name");

string edit\_manager = Console.ReadLine();

Console.WriteLine("Enter Updated Employees Count");

int edit\_count = int.Parse(Console.ReadLine());

UpdateDepartment(edit\_id, edit\_name, edit\_manager, edit\_count);

Console.WriteLine("Enter ID taht Department to delete");

int delete\_id = int.Parse(Console.ReadLine());

DeleteDepartment(delete\_id);

}

// CRUD operations

// C >Create / Insert

// R > Read

// U > Update

// D > Delete

// Get All Records

// SqlConnection comn = new SqlCOnnection();

// Sqlommand com = new Sqlommand("Select \* from bathes", connection);

// con.OPen();

// SlaadataReader reda = com.ExecuteRedaer();

// while (IReadOnlyCollection.Read()

// \{

// })}

//const.close();

static void GetDepartments()

{

// Read Operation

foreach (Department temp in db.Departments)

{

Console.WriteLine(temp.Id + " " + temp.Name + " " + temp.ManagerName + " " + temp.EmployeesCount);

}

}

static void InsertDepartment(int id, string name, string manager, int count)

{

// Create / Insert

Department department = new Department() { Id = id, Name = name, ManagerName = manager, EmployeesCount = count };

db.Departments.Add(department); // it will add only in collection

db.SaveChanges(); // SaveChanges send changes to server also

Console.WriteLine("Record Inserted");

}

static void UpdateDepartment(int id, string name, string manager, int count)

{

Department department = db.Departments.Where(x => x.Id == id).FirstOrDefault();

if(department != null)

{

foreach(Department temp in db.Departments)

{

if(temp.Id== id)

{

temp.Name = name;

temp.ManagerName = manager;

temp.EmployeesCount = count;

}

}

db.SaveChanges(); // SaveChanges send changes to server also

}

Console.WriteLine("Record Updated");

}

static void DeleteDepartment(int id)

{

Department department = db.Departments.Where(x => x.Id == id).FirstOrDefault();

if (department != null)

{

db.Departments.Remove(department);

db.SaveChanges(); // SaveChanges send changes to server also

}

Console.WriteLine("Record deleted");

}

}

}