Manage Staff And Department Details Using Fluent

**Scenario :**

Global Staff Selection Committee offers admission for various departments. They would like to create a software application to store the details of the departments provided by them and also maintain the details of staffs who have registered for various departments.

 Help them to create an application to store the details in the database.

Using C# and Entity Framework Fluent API approach, establish a one to many relationships between Department and Staff.

**Functionalities :**

·         Add department details to the database.

·         Add staff details to the database with the department name from the department table.

·         Retrieve the all staff details based on department name.

·         Retrieve the staff details based on the staff id.

**1. StaffContext.cs**

Implement the following classes

|  |  |
| --- | --- |
| **Class** | **Properties** |
| **Staff** | int Id |
| string Name |
| int Experience |
| double Salary |
| String DeptName |
| Department Department |

|  |  |
| --- | --- |
| **Class** | **Properties** |
| **Department** | int Code |
| String DeptName |
| ICollection<Staff> StaffList |
|  |
|  |

class StaffContext has the **DbContext**to establish connection with the SQLSERVER database.

*Additional Implementations in class StaffContext:*

Below should be implemented using Fluent API.

·         Department Entity MUST be mapped to **DepartmentDetail**table

·         **DeptName** should be the Primary key of **DepartmentDetail**table.

·         Staff Entity MUST be mapped to the **StaffDetail**table.

·         *Establish the following relationship between department and staff.*

              One-to-Many relationship between department and staff. One Department can have many staffs.

**Include navigation property at both ends.**

**a)**Department entity includes a collection navigation property of type **ICollection<Staff>**

**b)**Staff entity includes a reference navigation property of Department type with the foreign key property of **DeptName**

**2.  StaffUtility.cs**

|  |  |
| --- | --- |
| **class** | **Method** |
| **StaffUtility** | public Staff AddStaff(Staff staff) |
| public Department AddDepartment(Department department) |
| public List<Staff> GetStaffsList(string deptName) |
| public Staff GetStaffById(int Id) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Name** | **Argument** | **Return Type** | **Access Specifier** | **Responsibilities** |
| GetStaffById | int id | Staff | public | Find Staff details based on staff Id. Return the staff object. |
| GetStaffsList | string deptName | List<Staff> | public | Retrieve all the staffs based on the departmant name passed as argument. Return the list. |
| AddDepartment | Department   department | Department | public | This method should insert the given department details in the department object to the 'DepartmentDetail' table.  Return the stored department object. |
| AddStaff | Staff staff | Staff | public | This method should add the given Staff details in the staff object to the 'StaffDetail' table.   Return the stored staff object. |

**5. class Program**

Implement the Main method based on the Sample Input/Output given below.

*NOTE : 'Main' method is NOT tested by the platform.*

**Sample Input / Output :**

Enter the Department Code

1201

Enter the Department Name

HR

Department Details Inserted Successfully

Enter the Department Code

1202

Enter the Department Name

Admin

Department Details Inserted Successfully

Enter the Staff Name

John

Enter the Experience

10

Enter Salary

50000

Enter the Department Name

HR

Staff Details Inserted Successfully

Enter the Staff Name

Mathew

Enter the Department Name

Admin

Enter the Experience

6

Enter Salary

30000

Staff Details Inserted Successfully

Find Staff based on id:

**Enter Staff Id**

1

Name : John

Department : HR

Retrieve staff based on department name:

**Enter Department Name**

HR

John

1. Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StudentProject //DO NOT change the namespace name

{

class Program //DO NOT change the class name

{

static void Main(string[] args)

{

//Fill code here

Staff staff=new Staff();

Department department=new Department();

StaffUtility staffUtility=new StaffUtility();

Console.WriteLine("Enter the Department Code");

department.Code=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the Department Name");

department.DeptName=Console.ReadLine();

staffUtility.AddDepartment(department);

Console.WriteLine("Enter the Department Code");

department.Code=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the Department Name");

department.DeptName=Console.ReadLine();

staffUtility.AddDepartment(department);

Console.WriteLine("Enter the Staff Name")

staff.Name=Console.ReadLine();

Console.WriteLine("Enter the Experience");

staff.Experience=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Salary");

staff.Salary=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the Department Name");

staff.DeptName=Console.ReadLine();

staffUtility.AddStaff(staff);

Console.WriteLine("Enter the Staff Name")

staff.Name=Console.ReadLine();

Console.WriteLine("Enter the Experience");

staff.Experience=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter Salary");

staff.Salary=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the Department Name");

staff.DeptName=Console.ReadLine();

staffUtility.AddStaff(staff);

Console.WriteLine("Enter Staff Id");

int id=Convert.ToInt32(Console.ReadLine());

staff=staffUtility.GetStaffById(id);

Console.WriteLine("Find Staff based on id:");

Console.WriteLine("Name : {0}\nDepartment : {1}",staff.Name,staff.DeptName);

Console.WriteLine("Retrieve staff based on department name:");

Console.WriteLine("Enter Department Name");

string name=Console.ReadLine();

var l=staffUtility.GetStaffsList(name);

foreach(var item in l)

{

Console.WriteLine(item.Name);

}

}

}

}

2. StaffContext.cs

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

namespace StudentProject //DO NOT change the namespace name

{

public class StaffContext : DbContext //DO NOT change the class name

{

public StaffContext(): base("name=StaffContext"){}

//Implement virtual property for 'Staffs' and 'Departments' with required 'DbSet' declaration

public DbSet<Staff> Staffs { get; set; }

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

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

//Map Staff to StaffDetail table

modelBuilder.Entity<Staff>().ToTable("StaffDetail");

//Map Department to DepartmentDetail table

modelBuilder.Entity<Department>().ToTable("DepartmentDetail");

//Make 'DeptName' as primary key in Department Entity

modelBuilder.Entity<Department>().HasKey(t=>t.DeptName);

// configures one-to-many relationship as mentioned in the problem statement

modelBuilder.Entity<Department>().HasMany<Staff>(g=>g.StaffList).WithRequired(s=>s.Department).HasForeignKey(s=>s.DeptName);

}

}

public class Staff //DO NOT change the class name

{

public int Id { get; set; }

public string Name { get; set; }

public int Experience { get; set; }

public double Salary { get; set; }

// Add 2 properties

//1. Include a reference navigation property of Department type

public Department Department {get; set; }

//2. foreign key property of DeptName

public string DeptName{get; set; }

}

public class Department //DO NOT change the class name

{

public int Code { get; set; }

public string DeptName { get; set; }

//Include a collection navigation property of type ICollection<Staff>

public ICollection<Staff> StaffList{ get; set; }

}

}

3. StaffUtility.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace StudentProject //DO NOT change the namespace name

{

class StaffUtility //DO NOT change the class name

{

public Staff AddStaff(Staff staff) //DO NOT change the method signature

{

//fill code here

var sc= new StaffContext();

var s=sc.Staffs.Add(staff);

sc.SaveChanges();

Console.WriteLine("Staff Details Inserted Successfully");

return s;

}

public Department AddDepartment(Department department) //DO NOT change the method signature

{

//fill code here

var sc=new StaffContext();

var d=sc.Departments.Add(department);

sc.SaveChanges();

Console.WriteLine("Department Details Inserted Successfully");

return d;

}

public Staff GetStaffById(int Id) //DO NOT change the method signature

{

//fill code here

Staff s=null;

var sc=new StaffContext();

s=sc.Staffs.Find(Id);

return s;

}

public List<Staff> GetStaffsList(string deptName) //DO NOT change the method signature

{

//fill code here

var sc=new StaffContext();

var s=sc.Staffs.Where(c=>c.DeptName==deptName).Select(b=>b);

return s.ToList<Staff>();

}

}

}