FLUENT API

-another way to configure domain

-flexible better viewpoint wrt config

- 3 methods of configuration

- Entity

-Property

-Model

IMPLEMENTATION FLUENT API

-uses **DbModelbuilder** (map domain classes to db)

-override OnModelCreating method of DbContext object

- **DbModelbuilder class includes methods**

**-Modelwide configuration(database)**

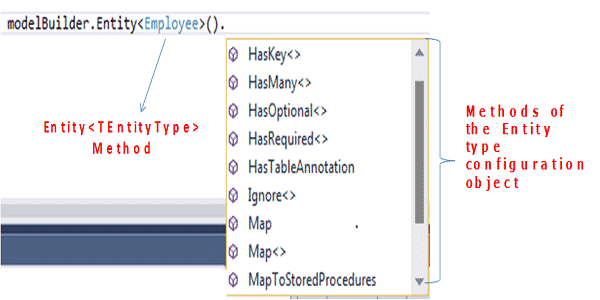
**-Entity Configuration(table)**

**-Propertyconfiguration**

ENTITY CONFIGURATION

-using method **Entity<TEntityType>**

-returns **entitytypeconfiguration** object to configure entities.



|  |  |
| --- | --- |
| Methods | Description |
| ToTable | Sets the table name for the entity type |
| HasKey | Configures the primary key for the selected entity type |
| HasForeignKey | Configures the Foreign key for the selected entity type |
| HasMany WithRequired | These methods are used to set relationships |
| HasOptional WithOptional |
| WithMany |
| WillCascadeOnDelete | Configure entities to turn off cascade delete |

PROPERTY CONFIGURATION

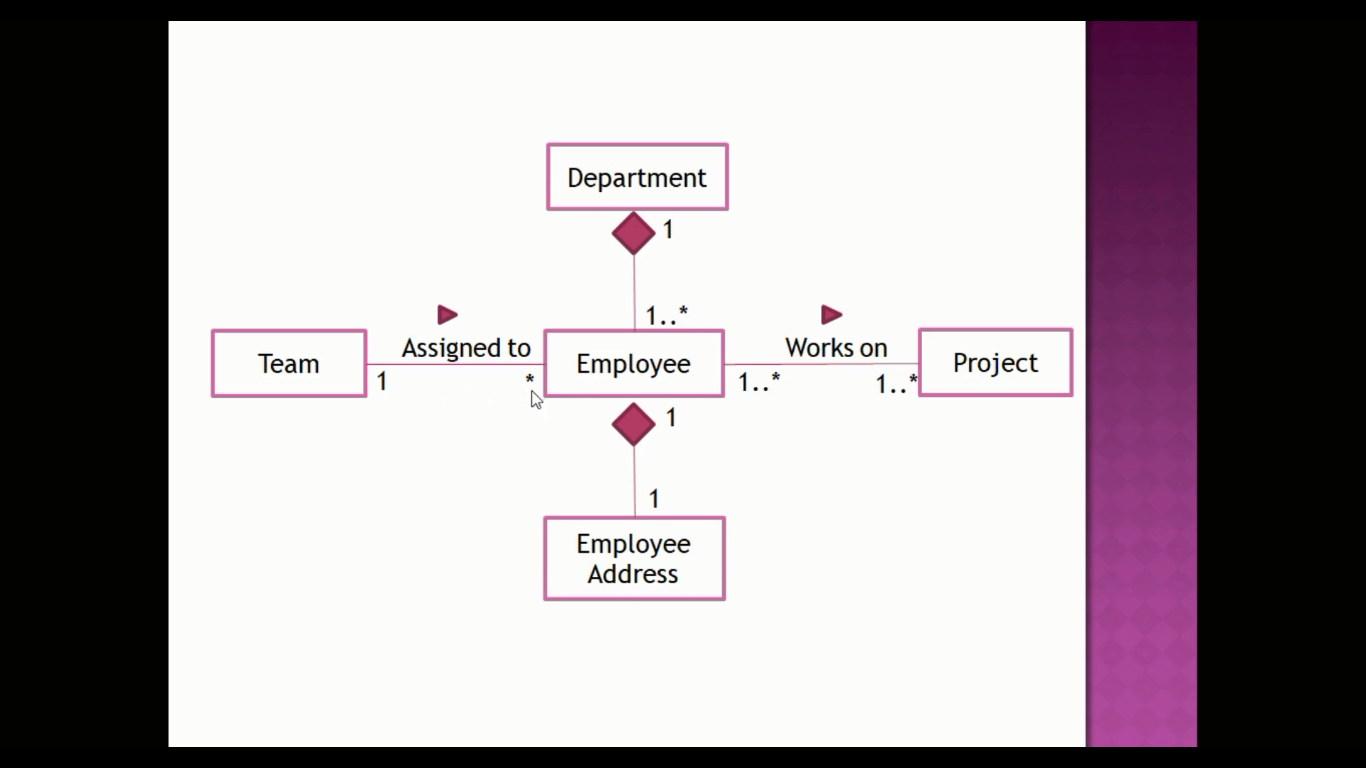
-**EntityTypeConfiguration** object returns property method.

-configures attributes of selected entity

-Configuration object has methods which can be used to configure

PROPERTY COM

|  |  |
| --- | --- |
| Methods | Description |
| HasColumnName | Configures database column name of the property |
| HasColumnOrder | Configures the order in which the column appears in the database table. |
| HasColumnType | Configures the database column data type of the property |
| HasDatabaseGeneratedType | Configures how values for the property are  generated by the database |
| HasMaxLength | Specifies the maximum length of the property. |
| IsOptional | Specifies the database column as nullable |
| IsReguired | Specifies the database column as non-nullable. |



Create a new project FluentAPI

App config

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

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />

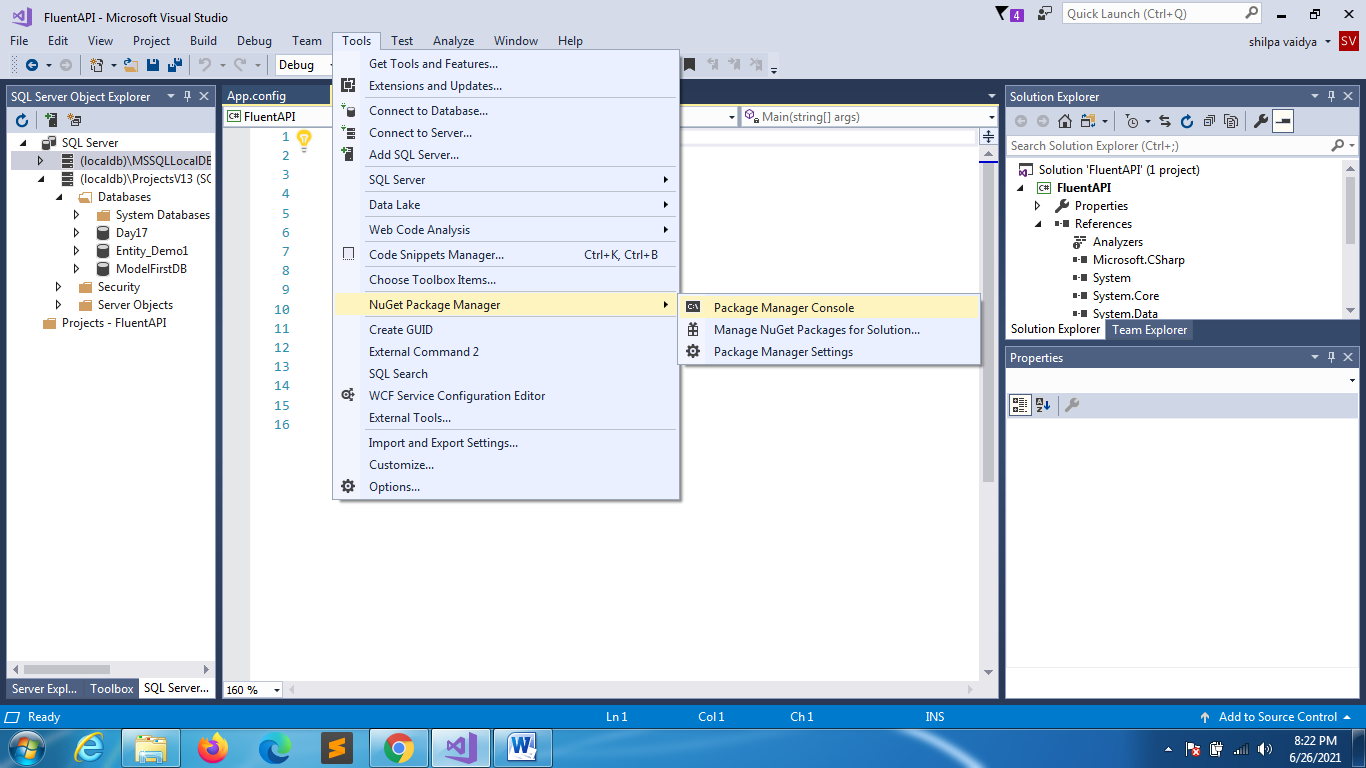
</startup>

<connectionStrings>

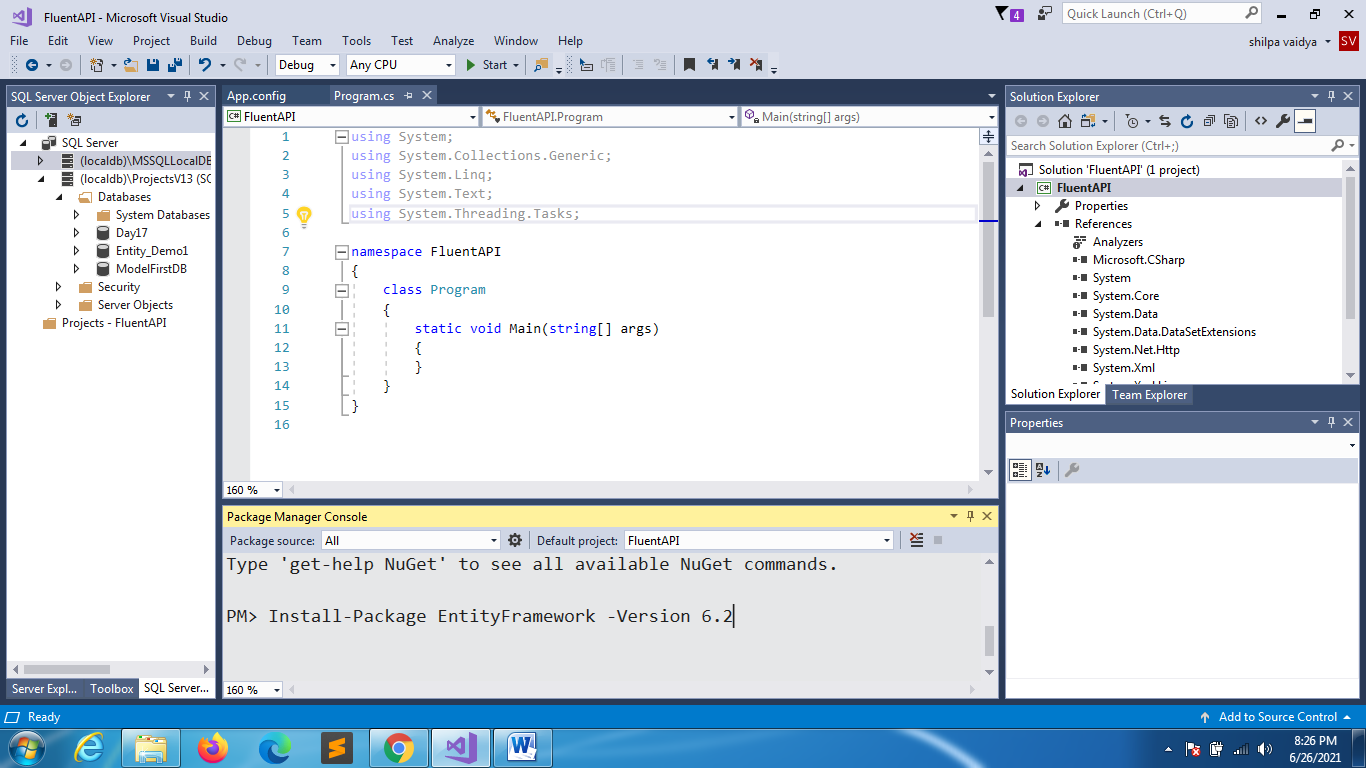
<add name="fluent" connectionString="Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=FluentAPIDB;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False" providerName="System.Data.SqlClient"/>

</connectionStrings>

</configuration>



Install-Package EntityFramework -Version 6.2



MyDbContext

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class MyDbContext:DbContext

{

public MyDbContext():base("name:fluent")

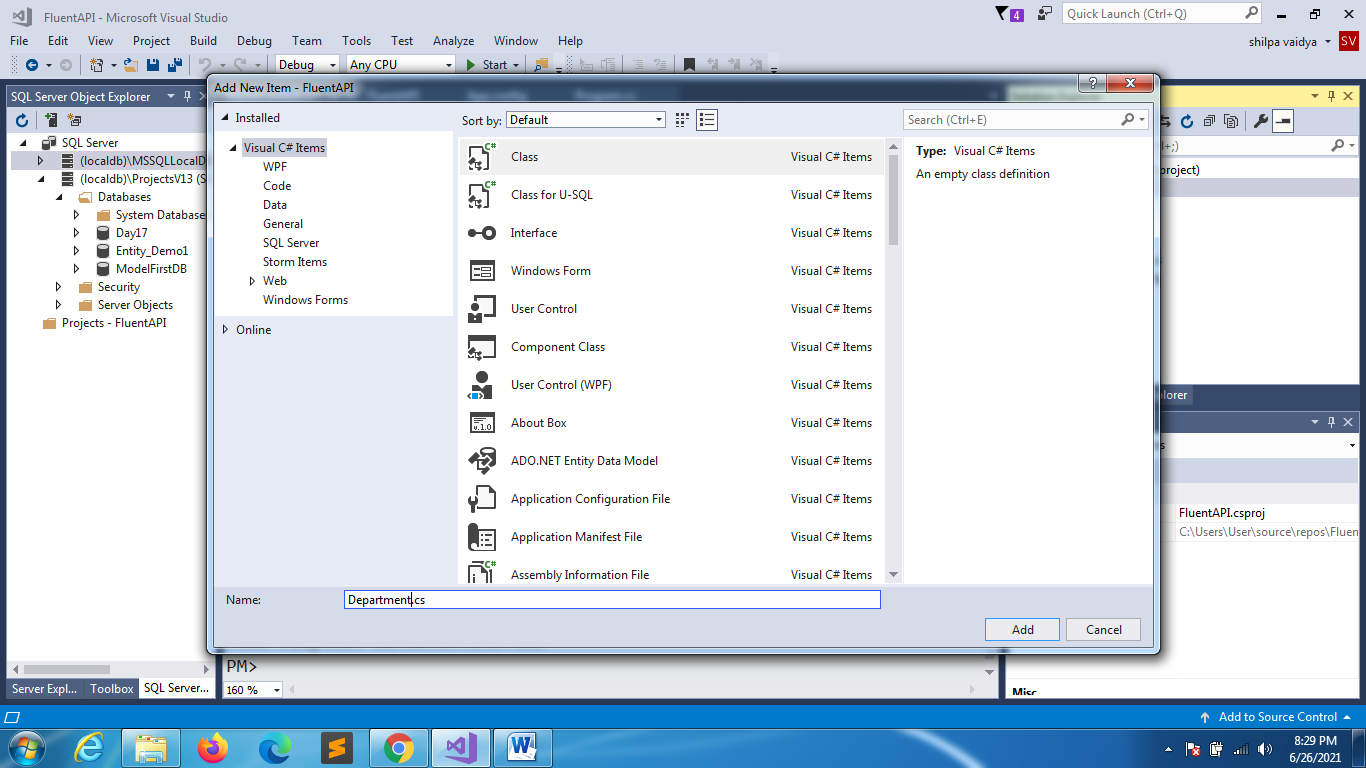
{

}

}

}

DEPARTMENT.CS



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Department

{

public int DeptId { get; set; }

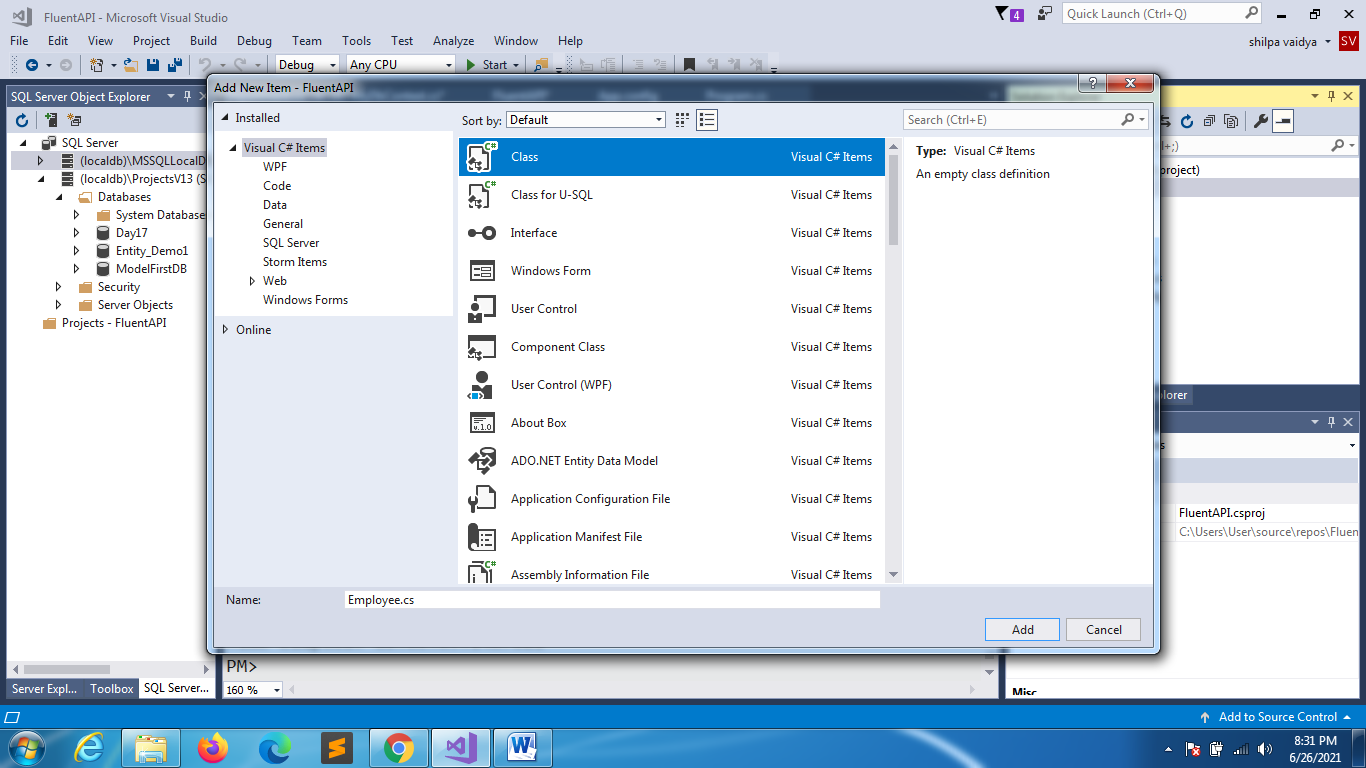
public string DeptName { get; set; }

//navigation properties

}

}

EMPLOYEE.CS



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Employee

{

public int EmployeeId { get; set; }

public string EmpName { get; set; }

public float Salary { get; set; }

public string Email { get; set; }

public string PrimaryContact { get; set; }

public string SecondaryContact { get; set; }

//navigatio properties

}

}

EMPLOYEEADDRESS.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class EmployeeAddress

{

public int EmployeeId { get; set; }

public string AddressLine1 { get; set; }

public string AddressLine2 { get; set; }

public string City { get; set; }

public string Zipcode { get; set; }

//navigation

}

}

TEAM.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Team

{

public int TeamId { get; set; }

public string TeamName { get; set; }

public int Size { get; set; }

}

}

PROJECT.CSusing System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Project

{

public int ProjectId { get; set; }

public string ProjName { get; set; }

public string ProjLanguage { get; set; }

public string Database { get; set; }

//navigation

}

}

NOW RELATIONSHIPS & NAVIGATION PROPERTIES

1.DEPARTMENT.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Department

{

public int DeptId { get; set; }

public string DeptName { get; set; }

//navigation properties

//one dept many employees and lazy loading so prop virtual

public virtual ICollection<Employee> Employees { get; set; }

}

}

EMPLOYEE.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Employee

{

public int EmployeeId { get; set; }

public string EmpName { get; set; }

public float Salary { get; set; }

public string Email { get; set; }

public string PrimaryContact { get; set; }

public string SecondaryContact { get; set; }

//navigatio properties

public virtual Department Department { get; set; }

}

}

EMPLOYEEADDRESS.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class EmployeeAddress

{

public int EmployeeId { get; set; }

public string AddressLine1 { get; set; }

public string AddressLine2 { get; set; }

public string City { get; set; }

public string Zipcode { get; set; }

//navigation

//one to one relation with emp

public Employee Employee { get; set; }

}

}

EMPLOYEE.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Employee

{

public int EmployeeId { get; set; }

public string EmpName { get; set; }

public float Salary { get; set; }

public string Email { get; set; }

public string PrimaryContact { get; set; }

public string SecondaryContact { get; set; }

//navigatio properties

public virtual Department Department { get; set; }

public virtual EmployeeAddress EmployeeAddress { get; set; }

}

}

TEAM.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Team

{

public int TeamId { get; set; }

public string TeamName { get; set; }

public int Size { get; set; }

//NAVIGATION

Public virtual ICollection<Employee> Employees { get; set; }

}

}

EMPLOYEE.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Employee

{

public int EmployeeId { get; set; }

public string EmpName { get; set; }

public float Salary { get; set; }

public string Email { get; set; }

public string PrimaryContact { get; set; }

public string SecondaryContact { get; set; }

//navigatio properties

public virtual Department Department { get; set; }

public virtual EmployeeAddress EmployeeAddress { get; set; }

//1 emp 1 team

public virtual Team Team { get; set; }

}

}

PROJECT.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Project

{

public int ProjectId { get; set; }

public string ProjName { get; set; }

public string ProjLanguage { get; set; }

public string Database { get; set; }

//navigation

//many to many rel

public virtual ICollection<Employee> Employees { get; set; }

}

}

EMPLOYEE.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class Employee

{

public int EmployeeId { get; set; }

public string EmpName { get; set; }

public float Salary { get; set; }

public string Email { get; set; }

public string PrimaryContact { get; set; }

public string SecondaryContact { get; set; }

//navigatio properties

public virtual Department Department { get; set; }

public virtual EmployeeAddress EmployeeAddress { get; set; }

//1 emp 1 team

public virtual Team Team { get; set; }

public virtual ICollection<Project> Projects { get; set; }

}

}

MYDBCONTEXT.CS

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FluentAPI

{

class MyDbContext:DbContext

{

public MyDbContext():base("name:fluent")

{

}

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

public virtual DbSet<Employee> Employees { get; set; }

public virtual DbSet<EmployeeAddress> EmployeeAddresses { get; set; }

public DbSet<Project> Projects { get; set; }

public DbSet<Team> Teams { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

//Totable -> configure/set table name in sql server

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

//HasKey (cause usually primary key will be either

//ID of DepartmentId in the class if it isnt so then we

//have to explicitly mention here

modelBuilder.Entity<Department>().HasKey(d=>d.DeptId);

}

}

}