Articles » Web Development » ASP.NET » Samples

Entity Framework Tutorial for Beginners

Code Help 2014, 7 Jul 2014

★★★★★ 4.77 (20 votes)

A simple tutorial for beginners to learn the basics of Entity Framework. It will teach you how to perform CRUD operations using EF.

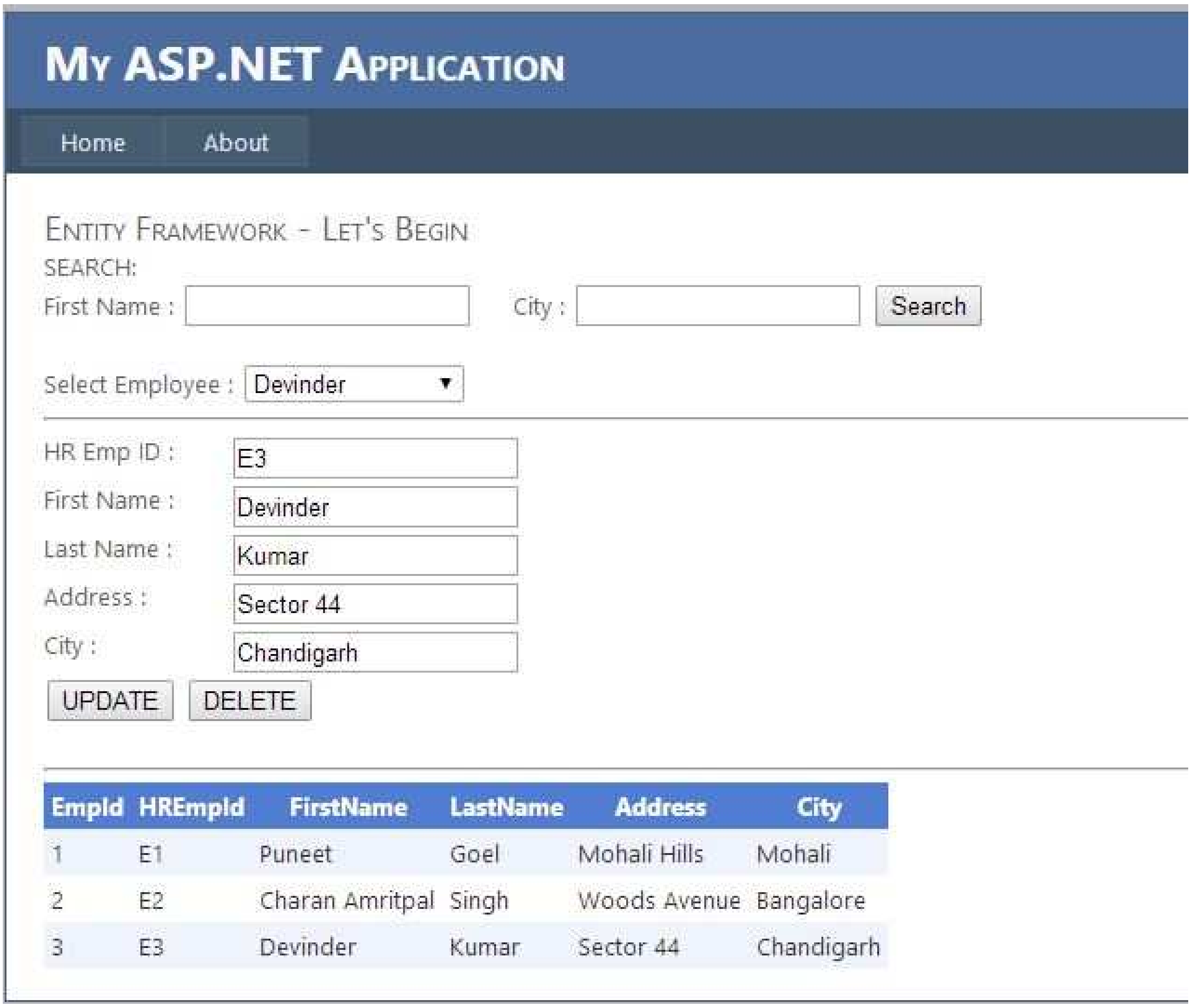
Download source - 174.8 KB

Introduction

In this article, we will learn the basics of Entity Framework by creating a sample application. This tip is for the programmers who have some experience in creating ASP.NET applications, but are new to Entity Framework. Here I will explain:

- How to create an Entity Data Model
- Perform CRUD operations
- Use Stored Procedures

Here is a quick view of the application that you are going to create:



Entity Framework - Let's Begin

SEARCH:

First Name :

City :

Select Employee : Devinder

HR Emp ID :	E3
First Name :	Devinder
Last Name :	Kumar
Address :	Sector 44
City :	Chandigarh

EmpId	HREmpId	FirstName	LastName	Address	City
1	E1	Puneet	Goel	Mohali Hills	Mohali
2	E2	Charan Amritpal	Singh	Woods Avenue	Bangalore
3	E3	Devinder	Kumar	Sector 44	Chandigarh

Background

There are many articles that will teach the basics of Entity Framework. But in this post, you are going to learn by creating a simple one page application.

What is Entity Framework?

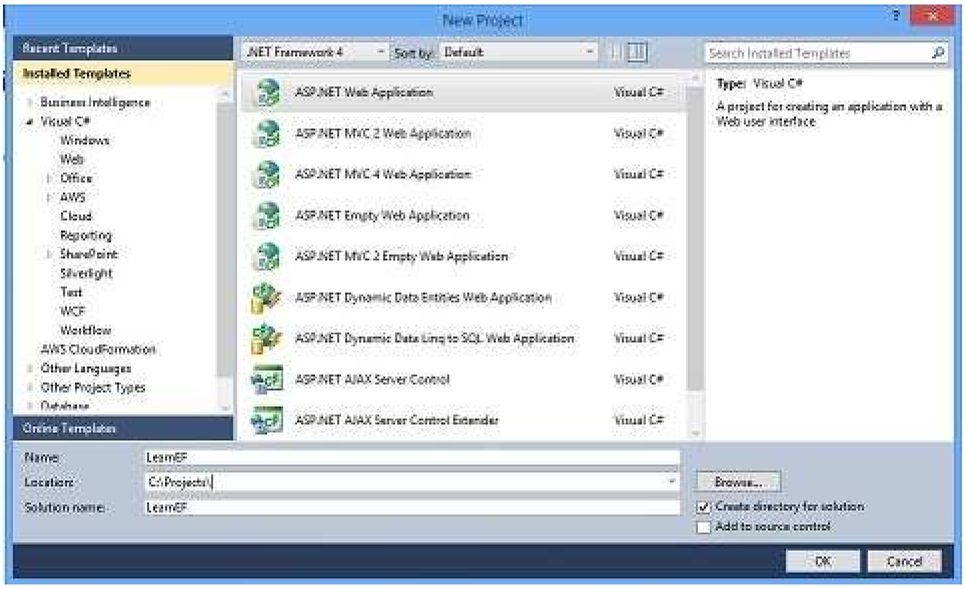
- Object/Relational Mapping (ORM) framework
- Work with database as domain-specific objects
- Retrieve and manipulate data as strongly typed objects

Using the Code

Let's walk through the application step by step.

Start a New Project

Start a new "ASP.NET Web Application" as shown below:


New Project
Recent Templates
Installed Templates
Business Intelligence
Visual C#
Windows
Web
Office
AWS
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
AWS CloudFormation
Other Languages
Other Project Types
Database
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
ASP.NET Web Application Visual C#
ASP.NET MVC 2 Web Application Visual C#
ASP.NET MVC 4 Web Application Visual C#
ASP.NET Empty Web Application Visual C#
ASP.NET MVC 2 Empty Web Application Visual C#
ASP.NET Dynamic Data Entities Web Application Visual C#
ASP.NET Dynamic Data Linq to SQL Web Application Visual C#
ASP.NET AJAX Server Control Visual C#
ASP.NET AJAX Server Control Extender Visual C#
Browse...
OK
Cancel

Type: Visual C#

A project for creating an application with a Web user interface

Name: LearnEF

Location: C:\Projects\

Solution name: LearnEF

☑ Create directory for solution

□ Add to source control

Create Database

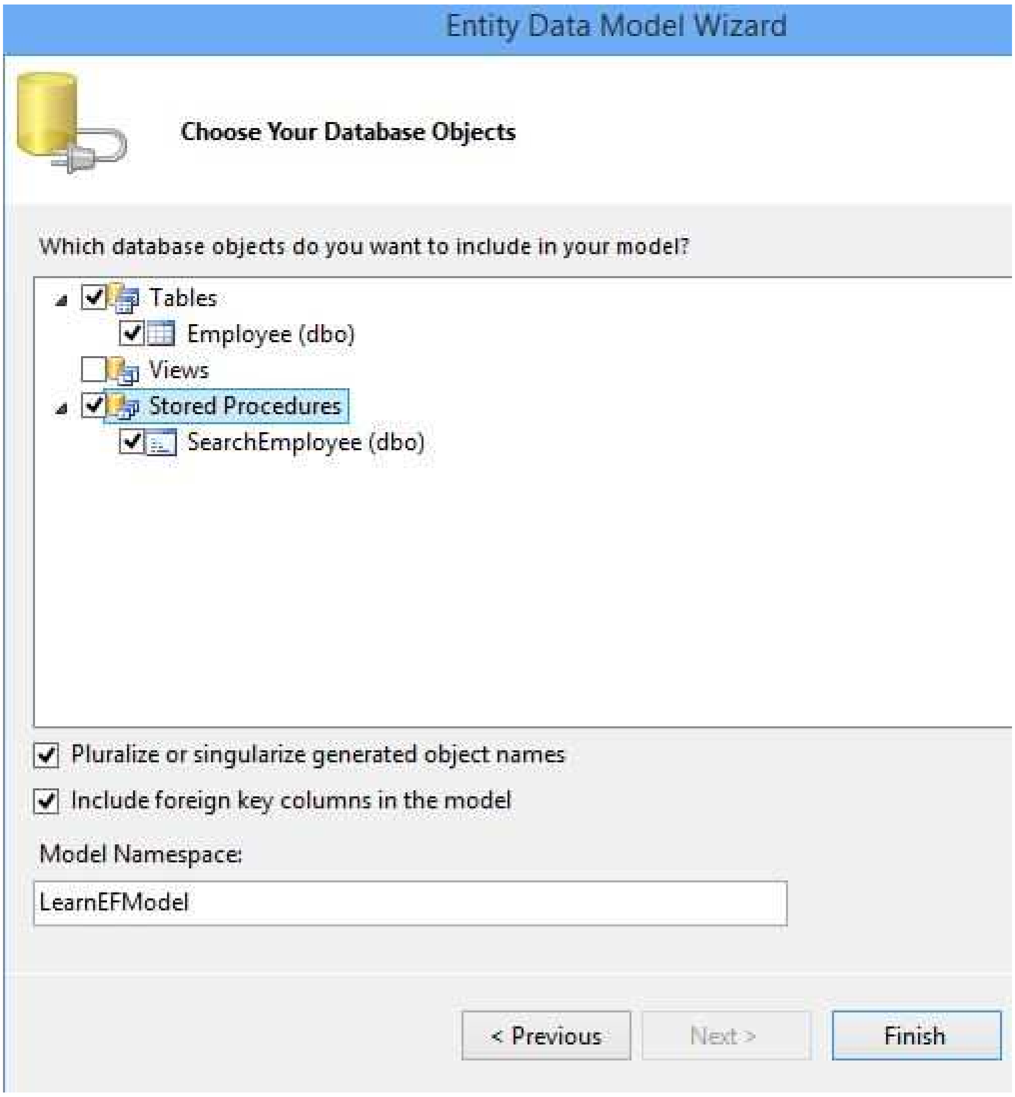
Here we are going to use a very simple database with name LearnEF with only one table. The table Employee will have a list of employees with basic information about each employee. You may use the following script to create the table and a stored procedure used in this application.

```
CREATE TABLE Employee
(
    EmpId        int NOT NULL IDENTITY(1,1) PRIMARY KEY,
    HREmpId        nvarchar(10),
    FirstName    nvarchar(30),
    LastName    nvarchar(30),
    Address        nvarchar(30),
    City        nvarchar(30)
)
GO

--SearchEmployee '', ''
CREATE PROCEDURE SearchEmployee
(
    @FirstName        varchar(30),
    @City            varchar(30)
)
AS
BEGIN
    DECLARE @query    nvarchar(1000)

    SET @query = 'SELECT * FROM Employee e'
    SET @query = @query + ' WHERE 1=1'

    IF @FirstName != ''
        SET @query = @query + ' AND FirstName LIKE ''' + @FirstName + '%'''

    IF @City != ''
        SET @query = @query + ' AND City LIKE ''' + @City + '%'''


    EXEC (@query)
END
GO
```

Create Entity Data Model

Now, let's create the entity data model. Here are the step-by-step details:

1. Right-click on the project name, select Add -> New Item.
2. Select the Data tab from left pane, then select ADO.NET Entity Data Model.
3. Name it as *LearnEF.edmx*. Click Add.
4. Select Generate from database in the Entity Data Model Wizard.
5. In the next window, set the connection properties for your database and click Next.
6. The next window will show you the objects in your database. Select the table `Employee` and the stored procedure `SearchEmployee`.



Entity Data Model Wizard

Choose Your Database Objects

Which database objects do you want to include in your model?

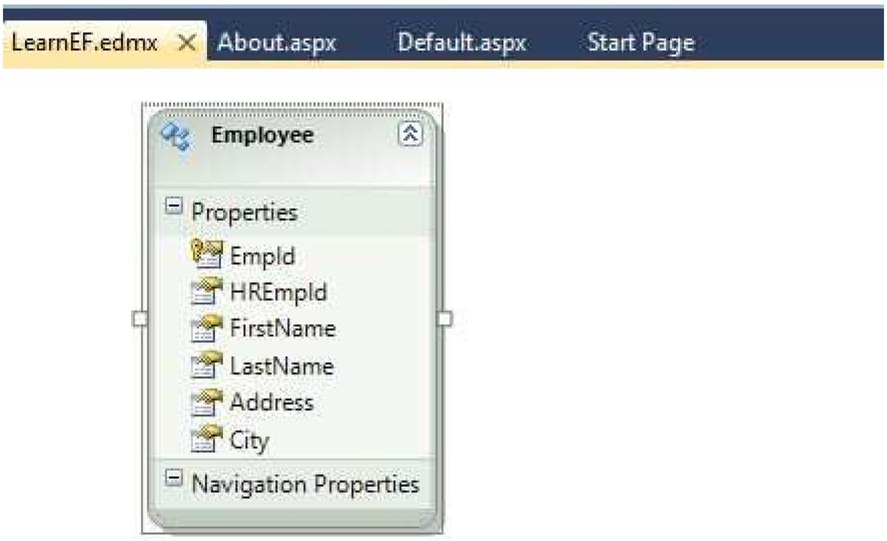
- ☑ Tables
  - ☑ Employee (dbo)
- □ Views
- ☑ Stored Procedures
  - ☑ SearchEmployee (dbo)

☑ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

Model Namespace: LearnEFModel

7. Click Finish. If you have completed the above steps successfully, you will see your EDM in design mode as shown below:



Insert a new record

To insert a new record:

1. Add an object of `Employee` class. e.g.

```
Employee objEmp = new Employee();
```

2. Set the value of the properties like:

```
objEmp.FirstName = txtFirstName.Text;
```

3. Add object to the collection in `ObjecContext` and call `SaveChanges`:

```
db.Employees.AddObject(objEmp);
db.SaveChanges();
```

Here is a complete code snippet:

```
LearnEFEntities db = new LearnEFEntities();

Employee objEmp = new Employee();
objEmp.HREmpId = txtHREmpId.Text;
objEmp.FirstName = txtFirstName.Text;
objEmp.LastName = txtLastName.Text;
objEmp.Address = txtAddress.Text;
objEmp.City = txtCity.Text;

db.Employees.AddObject(objEmp);
db.SaveChanges(); <span style="font-size: 9pt;"> </span>
```

Read from Employee table and populate GridView

```
LearnEFEntities db = new LearnEFEntities();

var empQuery = from emp in db.Employees
```

```
                select emp;
List<Employee> empList = empQuery.ToList();

ddlEmployee.DataSource = empList;
ddlEmployee.DataValueField = "EmpId";
ddlEmployee.DataTextField = "FirstName";
ddlEmployee.DataBind();

ddlEmployee.Items.Insert(0, new ListItem("--Add New--", "0"));

//bind grid
GridView1.DataSource = empList;
GridView1.DataBind();
```

Update an employee record

```
LearnEFEntities db = new LearnEFEntities();

                var empQuery = from emp in db.Employees
                               where emp.EmpId == empId
                               select emp;
                Employee objEmp = empQuery.Single();

                objEmp.HREmpId = txtHREmpId.Text;
                objEmp.FirstName = txtFirstName.Text;
                objEmp.LastName = txtLastName.Text;
                objEmp.Address = txtAddress.Text;
                objEmp.City = txtCity.Text;

                db.SaveChanges();
```

How to Delete a record

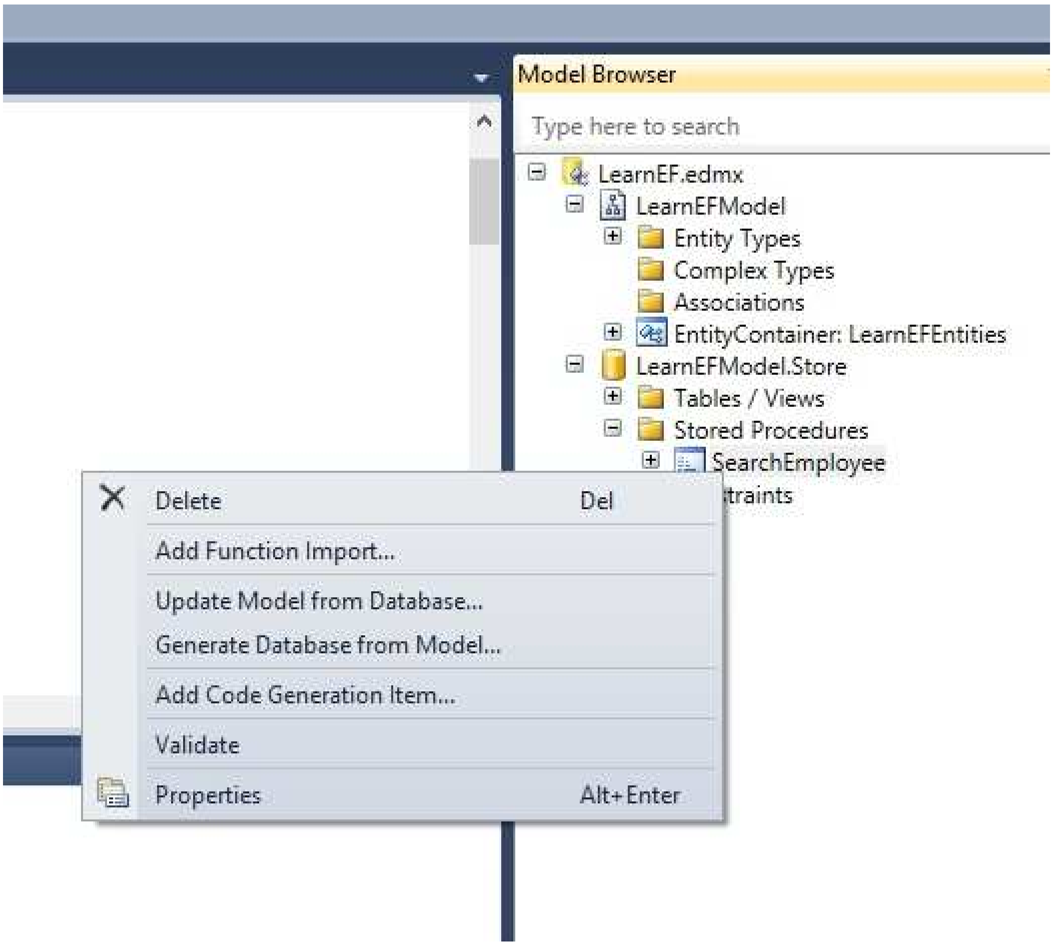
The following code will explain the steps to delete a record:

```
LearnEFEntities db = new LearnEFEntities();

//create a new object using the value of EmpId
Employee objEmp = new Employee() { EmpId = empId };

//attach and delete object
db.Employees.Attach(objEmp);
db.Employees.DeleteObject(objEmp);

//save changes
db.SaveChanges();
```

How to use Stored Procedures in Entity Framework

We will add the search functionality using a stored procedure. For this:

1. Double click *LearnEF.edmx*
2. Right click in the main pane, then select Model Browser.
3. Expand Stored Procedure node in model browser, right click `SearchEmployee` stored procedure, then click Add Function Import.



4. Set "Returns a collection of" value to Entities and select `Employee` from drop-down.
5. Click OK and now we are ready to use this stored procedure.

The statement to use this stored procedure will be :

```
List<Employee> empList = db.SearchEmployee(txtSrchFirstName.Text, txtSrchCity.Text).ToList();
```

The following code snippet can be used to search employees and bind a grid:

```
List<Employee> empList = db.SearchEmployee(txtSrchFirstName.Text, txtSrchCity.Text).ToList();

//bind grid
GridView1.DataSource = empList;
GridView1.DataBind();
```

License

Share

About the Author

No Biography provided

Code Help 2014

Software Developer (Senior)

India

You may also be interested in...

Enterprise Integration Patterns Flash Cards

IDC: How Red Hat's JBoss Enterprise Application Platform is Extending Business Value

Comments and Discussions

12 messages have been posted for this article Visit **http://www.codeproject.com/Tips/739164/Entity-Framework-Tutorial-for-Beginners** to post and view comments on this article, or click **here** to get a print view with messages.

Permalink | Advertise | Privacy | Mobile
Web02 | 2.8.140926.1 | Last Updated 7 Jul 2014