.net POCO Classes in Entity FrameWork

Munesh Sharma

Updated date May 04 2017 331.6k 11 8

POCO

A Plain Old CLR Objects (POCO) is a class, which doesn't depend on any framework-specific base class. It is like any other normal .NET class. Due to this, they are called Plain Old CLR Objects. These POCO entities (also known as persistence-ignorant objects) support most of the same LINQ queries as Entity Object derived entities. These classes (POCO classes) implements only the domain business logic of the Application.

Some developers use Data Transfer Objects (DTOs) with the classes to pass the data between the layers because POCOs are also used to pass the data between the layers, but they become heavy. Hence they use DTOs, which are also theclasses.

The main difference between DTO and POCO is that DTOs do not contain any methods. They only contain public members. Thus, sending the data, using a DTO is easy because they are lightweight objects.

The code given below defines a POCO class.

classes.

To create POCO classes, we first need to disable auto create classes or auto create code generation, which generates Context classes entity code in Model1.designer.cs. To disable this code generation, right-click on model1.edmx (ADO.NET data modal) and click property Now, you will see the value of "Custom Tool" as **EntityModelCodeGenerator** and you need to remove this value.

After removing the value Custom tool, you will see that in modal1.edmx, there is no Model1.designer class. Now, we must create properties (Context and Entities), so for this, we need to create POCOs classes.

Now, double-click on Modal1.edmx and right-click on the designer surface and click on the code generation Items. A screen will open from where you need to select ADO.NET POCO Entity Generator and click Add.

After clickinig Add button, you will see 2 classes, where one is **modal1.context.tt** and another is **modal1.tt.**

Model1.Context.tt is a context file and Model1.tt is an entity file. You can modify this file, if you want to generate your template. The Model1.Context.cs file has a context class and .cs files under Model1.tt are the entity classes.

Entity classes have all the properties as Virtual. In other words, these entities fulfill the requirements of POCO Proxy entities. These entities can be used as POCO entities or POCO Proxy entities. By default, it will behave as POCO Proxy entities, but you can disable proxy creation by setting a property "ObjectContext.ContextOptions.ProxyCreationEnabled = false".

Note

- If you are not able to see ADO.NET POCO Entity Generator, then you must install NuGet Package Library.

Code First Approach in Entity Framework

Entity FrameWork POCO Classes POCO Classes in Entity FrameWork

Munesh Sharma TOP 500

Hello everyone! I'm Munesh (Blogger, Writer, Softwre Developer), I am graduate in Information Technology from Rajasthan Technical University, India. My skills are:- C,Asp.net, C#,Ado.net, SQL SERVER, MVC,web services... Read more

http://www.dotnetoffice.com/

165 3.8m 3

View Previous Comments

8 11

Type your comment here and press Enter Key (Minimum 10 characters)

Super article Munesh, this is very helpful to me.

Bhavesh Jadav

• 979 • 1.1k • 4.4k

Nov 15

0 0

1901 3 0

Hi. What VS-version is this for? I cannot see the poco generator, so wondering what nuget packag e t needs to be installed? Thanks for any help.

Gert Hauan

1901 3 0

Jan 22, 2019

0 0 Reply

Good job, Munesh! The only unfortunate thing is: the photos in the article can not be displayed on my browser(both Chrome and Fire

1902 2 0

0 0 Reply

Good Article.

Vetri Chelvan Indrajith

1453 452 154.2k

Aug 09, 2016

0 0 Reply

Good one

Sr Karthiga

164 12.1k 969.1k

May 03, 2016

0 0 Reply

Nice Work

Sharath Gowda

1870 34 0

Apr 29, 2016

0 0 Reply

Hi Sam i got only this link for Old CLR Object " https://msdn.microsoft.com/en-us/library/vstudio/dd456853(v=vs.100).aspx" .. not a such nice link

Munesh Sharma

165 12.1k 3.8m

Mar 03, 2015

0 0 Reply

Nice one..

Shaili Dashora

385 5.1k 1.8m

Mar 03, 2015

0 0 Reply

Good Show...

Rahul Saxena

Mar 03

FEATURED ARTICLES

Create A Drill Through Button In Power BI

Azure Kubernetes Service Architecture

Number Series Codes using C#

Pagination in ReactJS Using Web API and SQL

View All

TRENDING UP

01 ASP.NET Core Web API - Creating And Validating JWT (JSON Web Token)

02 Understanding ASP.NET Core 3 Razor Pages Project Files

03 Performing Update and Delete Operations in ASP.NET Core 3 Razor Pages Using Microsoft SQL Server

04 Putting The Fun In C# Local Functions

05 Creating a Model and Database in ASP.NET Core Razor Pages Using Entity Framework Core

07 Inserting Data into SQL Server Database Using ASP.NET Core 3 Razor Pages

08 Implement JWT In ASP.NET Core 3.1

09 Create Docker Image for Simple Web Application and Hosting using Visual Studio .NET Core

10 ASP.NET Core API Analyzer

View All