

# What is Entity Framework fluent api?

▲ I keep hearing about the Entity Framework fluent-api but I am struggling to find a good reference on this. What is it?

52 ▼ We use the entity framework and the modeling tool provided. Is that all that is? Or is it something different?

★ Similarly, if it's not too broad a question, what is POCO? I know it stands for Plain Old CLR Objects, but what does that mean to me as somebody who uses EF already with the designer model tool? If that question is too vague then please ignore it. I'm just learning here and any information you are willing to provide is helpful.

24

c# entity-framework orm entity-framework-4 ef-fluent-api

edited Nov 28 '17 at 20:46



abatishchev

71.5k 70 269 399

asked Jun 13 '11 at 15:15



Chev

36.9k 49 188 298

## 5 Answers

▲ Entity Framework 4.1 introduces the code first approach of writing database models. This is also called POCO (Plain Old CLR Objects). The idea is that you can build your database from these classes, rather than building the database first and creating a model from that.

56 ▼ There are tons of good blog articles and MSDN documentation on this. A good place to start would be

<http://blogs.msdn.com/b/adonet/archive/2010/12/14/ef-feature-ctp5-fluent-api-samples.aspx>

✓ <http://weblogs.asp.net/scottgu/archive/2010/12/08/announcing-entity-framework-code-first-ctp5-release.aspx>

<http://weblogs.asp.net/manavi/archive/2011/03/27/associations-in-ef-4-1-code-first-part-1-introduction-and-basic-concepts.aspx>

Regards the fluent API, this is basically using the EF classes to build your database e.g.:

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



```
public class Category
{
    [Key]
    public int CategoryCode { get; set; }
}
```

The `[Key]` attribute comes from [Data Annotations](#)

edited Nov 2 '12 at 14:01



Crwth

1,092

1

8

13

answered Jun 13 '11 at 15:23



Jason Evans

24.5k

12

76

135

- 2 Jason, Am I missing something big if I decided to use Data Annotations instead of fluent api. I'm just starting with MVC and this topic is a bigger curve for me. – [Crismogram](#) Mar 18 '15 at 2:14 ✎
- 5 I don't believe you're missing out. But I would say that using the fluent API means you can put all the database setup in one method, rather than spreading that setup around in attributes that are associated with different classes. So it's a case of which way you prefer really. – [Jason Evans](#) Mar 18 '15 at 8:18

@JasonEvans good response, in the same way, I want to know what is the best tools to take all the controls of our database, fluent-api or data annotations? especially when we work on a big project with a lot of database structure changes? – [Thamer belfkih](#) Sep 29 '18 at 10:48

POCO stands for Plain Old CLR Object.

5

[Article](#) on Fluent API.

answered Jun 13 '11 at 15:22



Xaisoft

21.1k

79

254

396

You can also check the Code First Fluent API section on MSDN here <http://msdn.microsoft.com/en-us/library/hh295844>

Join **Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



0

Responding to your POCO question: in the application I'm currently working on I'm using POCO's to pass data to my Silverlight front end (EF just wasn't cutting it). Essentially, I use the entities that the EF modeler created, massage them into a serializable-friendly version, and then send them back and forth over the wire. POCO's are there to provide a layer of abstraction when needed. I think of it as an adaption of the DAO pattern to serialization, instead of using it for DB access like the DAO pattern normally does.

answered Jun 13 '11 at 15:27

**Eric Andres** ♦**2,967** 2 19 37

0

See the paragraph on POCO classes in <http://www.asp.net/entity-framework/tutorials/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application>

Essentially, in the context of EF, POCO classes are entity classes that do not inherit from the Entity Framework `EntityType` class (which is what you get by default in Database First or Model First). As one of the other answers mentions, this makes it easier to serialize the objects, but also some development and automated testing methodologies prefer to work with objects that have no reference to the Entity Framework.

answered Jun 15 '11 at 0:57

**tdykstra****4,987** 2 16 17

Join **Stack Overflow** to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)