

Why is EF returning a proxy class instead of the actual entity?

Asked 8 years ago Active 7 months ago Viewed 30k times



31



10



I'm having trouble with entity framework returning Proxies when I want the actual entity class. The first time I run my code everything runs properly (no proxies), but every iteration afterwards one of my DbSet's always returns proxies instead of the actual type.

I dispose of the context after every iteration, so I don't understand why the first time through it works, and every time after doesn't.

My code fails on this line. All my POCOs have the Table attribute set, but because it is returning a proxy class there is no table attribute.

```
TableAttribute attrib = (TableAttribute)attributes.Single();
```

Is there some behind the scenes static magic in the DbContext that lives after I destroy the object?

I move my objects into memory using the following

```
MajorClasses = ctx.MajorClasses.ToArray();
```

I also tried

```
MajorClasses = ctx.MajorClasses.AsNoTracking().ToArray();
```

In my OnModelCreating I have the following set

```
base.Configuration.ProxyCreationEnabled = false;  
base.Configuration.LazyLoadingEnabled = false;
```

c# entity-framework

edited Sep 13 '17 at 16:00



JohnOsborne

863 ● 9 ● 25

asked Feb 29 '12 at 15:03



Malcolm O'Hare

4,455 ● 3 ● 25 ● 48

6 Answers

Active

Oldest

Votes



47



You can set `ObjectContext.ContextOptions.ProxyCreationEnabled` to false. This will prevent you from using some of EFs fancy features like lazy loading and I believe change tracking.

As far as your app cares, it should be able to treat the proxies just like the types they represent. Is there a specific issue you are having?

Edit

We have some code that requires the POCO type instead of the proxy type and we do the following to detect if the current type is a proxy.

```
if (entityType.BaseType != null && entityType.Namespace ==
    "System.Data.Entity.DynamicProxies")
{
    entityType = entityType.BaseType;
}
```

edited Feb 29 '12 at 15:30

answered Feb 29 '12 at 15:18



cadrell0

15.3k ● 3 ● 46 ● 67

I added more detail to my question. I did have this setting set and no my application can't treat the proxies like the POCO. – [Malcolm O'Hare](#) Feb 29 '12 at 15:25

Its in the updated question. I reflect the class to get attributes from it. – [Malcolm O'Hare](#) Feb 29 '12 at 15:27

@MalcolmO'Hare I've posted the code we use to get the POCO type from the proxy type. – [cadrell0](#) Feb 29 '12 at 15:31

Perfect solution, works wonderfully. Still wondering whats causing the proxy instead of the baseType but if it works I don't really care. – [Malcolm O'Hare](#) Feb 29 '12 at 15:52

3 @Matthew I never said that it did. It does however give you a Type object that represents the POCO class. From here you can use reflection to get the attributes in question. If my answer wasn't able to solve your problem, post a new question and I'll see if I can answer it. – [cadrell0](#) Aug 30 '12 at

16:39

▲ To turn off proxy creation in Entity Framework 5 you can use the following,

6

```
_dbContext.Configuration.ProxyCreationEnabled = false;
```

▼ Simply set this property once before using the context to pull data.



answered Oct 31 '13 at 23:00

**BizarroDavid**

726 ● 8 ● 8

2

▲ By default, EF uses [Change Tracking](#) and uses an in-memory cache of all entities. You can use different Merge Options when working with EF. By default, EF 4.1 is set to AppendOnly Merge Option. As I understand, this means that if you have already queried an entity, subsequent queries will get the entity from the cache (if there are no detected changes in the database). So you might be seeing the cached entity coming back.

▼ In EF 4.1, you can use NoTracking Merge Option. This will go to the database for every call.



answered Feb 29 '12 at 15:10

**Davin Tryon**

58.4k ● 10 ● 131 ● 121

On every iteration of my program (its polling a database, when a record is inserted do work on another database) I create a new context and load all my data into memory. When all my work is done all the loaded data + the context is disposed of. If I run the same export twice, the second time it runs some records are returned as proxies, which didn't happen the first time it ran. Also I'm using 4.3 now (upgraded to see if it fixed the problem). –

[Malcolm O'Hare](#) Feb 29 '12 at 15:14

Yeah, it does sound strange that you would get a proxy object based on that. You could set to NoTracking just to rule out the cache, though. –

[Davin Tryon](#) Feb 29 '12 at 15:16

1 @MalcolmO'Hare Are the proxies virtual properties of other entities? Maybe you are just seeing EF lazy loading. If you don't use (load) the entity set up as virtual then it might appear as a proxy. – [Davin Tryon](#) Feb 29 '12 at 15:16

▲ In EF 6.1.3 you can get the right type using

0

```
using (var context = new BloggingContext()) {  
    var blog = context.Blogs.Find(1);  
    var entityType = ObjectContext.GetObjectType(blog.GetType());  
}
```



Note that if the type passed to `GetObjectType` is an instance of an entity type that is not a proxy type then the type of entity is still returned. This means you can always use this method to get the actual entity type without any other checking to see if the type is a proxy type or not.

From [MSDN](#)

answered Feb 14 '17 at 15:52



[Kautsky Lozano](#)

482 ● 8 ● 15

3 He wants the actual entity, instead of it's type. – [Tito](#) Jul 17 '17 at 8:22

This still returns the dynamic proxy type anyway!? – [jamheadart](#) Feb 5 at 16:18

@jamheadart This return the actual entity type and not the proxy type – [Kautsky Lozano](#) Feb 7 at 8:58

simple solution is you are missing some object that must be included and also do this one before getting values

0

```
_dbContext.Configuration.ProxyCreationEnabled = false;
```



edited Jul 15 '19 at 9:47



[David](#)

18.5k ● 9 ● 48 ● 74

answered Jul 15 '19 at 9:26



[Hassan Raza](#)

11 ● 3

In my case this issue was fixed by setting `Lazy Loading Enabled` to `false`.

0

1. Open the .edmx (diagram)
2. Hit F4 to bring up the properties
3. Set `Lazy Loading Enabled` to `false`



4. Save and rebuild

answered Aug 13 '19 at 16:01



[cdslin](#)

651 ● 8 ● 29