

# Entity Framework auto generate GUID

I am new to EF so here goes. I have a class which contains the following

33

```
public class EmailTemplate
{
    public Guid Id { get; set; }

    [MaxLength(2000)]
    public string Html { get; set; }
}
```

3

Here is my mapping class

```
class EmailMapper : EntityTypeConfiguration<EmailTemplate>
{
    public EmailMapper()
    {
        ToTable("EmailTemplate");

        HasKey(c => c.Id);
        Property(c =>
c.Id).HasDatabaseGeneratedOption(DatabaseGeneratedOption.Identity);
        Property(c => c.Id).IsRequired();
    }
}
```

I am trying to call `DbContext.SaveChanges()` , but I get the following error :

Exception Details: System.Data.SqlClient.SqlException: Cannot insert the value NULL into column 'Id', table 'AutoSendConnection.dbo.EmailTemplates'; column does not allow nulls. INSERT fails.

What am i doing wrong? Why won't EF auto create a unique GUID?

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

Sign up with email

 Sign up with Google

Sign up with Facebook





## 7 Answers



Just decorate the Id field on your EmailTemplate class as below and SQL Server will automatically generate the value on insert.

48



```
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]  
[Key]  
public Guid Id { get; set; }
```

You can also remove your Mapper class as it's no longer needed.

edited Aug 2 '14 at 19:07

answered Aug 2 '14 at 19:01



Mark

1,258 8 10

Actually, this is the same as what OP already did in their code. Although this is a workaround which fixes the problem, it doesn't answer the original problem. – [Alisson](#) May 23 '17 at 1:42

If you use CodeFirst, EF might not detect this change. You can delete your table and add a new migration. – [Virgar Poulsen](#) Aug 19 '17 at 6:09

No default sql value set in the migration generated so this doesn't work (even when creating the table from scratch). – [War](#) May 1 at 11:30

This answer works for EF Core only ... EF6 users will find this problematic. – [War](#) May 31 at 17:18



If using .Net core then this should work for you ...

16

Use fluent API

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

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

Sign up with email



Sign up with Google

Sign up with Facebook



```
modelBuilder.Entity<Student>().Property(p => p.StudentID)
    .HasDatabaseGeneratedOption(DatabaseGeneratedOption.Identity);
```

Here is a more comprehensive [Cheat Sheet for entity framework](#)

edited May 1 at 11:51



War

6,347

2

36

82

answered Mar 28 '17 at 9:38



Mohsen Tabareh

468

4

7

What is Node? Here it says that it an assembly may be missing. I'm using EF 6.1. Edited: It is the Model. – [acido](#) Feb 8 '18 at 19:20

I'm getting an error with the "HasDefaultValueSql" it does not exist on EF 6.1 anymore. I'm trying to look for the alternative – [acido](#) Feb 8 '18 at 19:25

1 @acido Node is an entity in EF, can be any db table – [Mohsen Tabareh](#) Feb 9 '18 at 20:18

2 this won't work if you are using MSSql and then using an in memory Sql Lite for testing. – [Parv Sharma](#) Mar 23 '18 at 6:38

HasDefaultValueSql() doesn't seem to exist (where are you getting that from?) and the second option doesn't work, the migration generated doesn't include the default sql setting to the primary key as it should. – [War](#) May 1 at 11:31

Set the default sql value of the field to 'newsequentialid()' in the mapping configuration.

4

edited Aug 2 '14 at 12:12

answered Aug 2 '14 at 11:57



Liviu M.

5,585

1

34

60

Please see my mapping class in the amended question, Where would i put the newsequentialid() ? I am trying to do this via code first and just thought the EF would handle it? – [user2859298](#) Aug 2 '14 at 12:47

This shows an example: [blog.jongallant.com/2013/04/...](http://blog.jongallant.com/2013/04/...) – [Liviu M.](#) Aug 2 '14 at 13:05

You can also set Default Value of ID as NewID() in Sql Server itself and pass the GUID as null.

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

Sign up with email



Sign up with Google

Sign up with Facebook





Moons

2,666 2 39 77

- 1 Sequential ID causes less fragmentation of the indexes, hence better query performance – [Livi M.](#) Aug 2 '14 at 12:43

Even if u set Default value of ID as NewID() in SQL server, U still need to modify your .edmx file if StoreGeneratedPattern property not yet set. ref: [developerhandbook.com/entity-framework/...](http://developerhandbook.com/entity-framework/...) – [Chinh Phan](#) Mar 15 '16 at 7:16

@ChinhPhan In standard cases i dont think we need to change any XML. I use wizard to create EDMX files and everything worked fine. – [Moons](#) Mar 17 '16 at 13:12

I prefer to leave the database to generate the id automatically for example the following schema:

```
1 CREATE TABLE [dbo].[MyTable](  
    [MyId] [uniqueidentifier] NOT NULL CONSTRAINT [DF_Booking_BookingId] DEFAULT  
    (newsequentialid())  
)
```

Then in the code first mapping I specify the following to tell Entity Framework that the database will take care of generating the value on insert.

```
Property(a =>  
a.MyId).IsRequired().HasDatabaseGeneratedOption(DatabaseGeneratedOption.Identity);
```

answered May 28 '18 at 7:18



John

2,806 1 31 51

- 1 With code first we don't write the SQL the migration does, and the migration code you show there doesn't result in the SQL you've shown, see my answer for my details. – [War](#) May 1 at 11:49

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

Sign up with email



Sign up with Google

Sign up with Facebook



<https://github.com/aspnet/EntityFramework6/issues/762>

... for some reason the EF dev team seem to think that this is "working by design" and repeatedly close tickets questioning this "bug".

### The EF team explanation

For some reason they seem to think that "generating guids in SQL is considered not best practice and that to ensure the keys are available immediately we should generating the keys in the app code".

The issue here of course is that highly populated tables run the risk of you taking further business actions consuming an invalid key.

In my case this could break some extremely complex multi server DTC transactions so I don't believe the advice from MS to be correct.

### My answer (which actually works)

In short, I solved this by "manually hacking" the generated migration after generating it ...

[EF code first migrations, DB generated guid keys](#)

To quote the other question the answer is ...

Generate the migration script as you normally would putting both attribs on the key property like this ...

```
public class Foo
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public Guid Id { get; set; }
}
```

.. declaratively speaking the entity is now correct.

The migration it will generate will look something like:

```
CreateTable(
    "dbo.Foos",
    c => new
```

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

Sign up with email



Sign up with Google

Sign up with Facebook



... I have not been able to pin down why, but in some situations this will work and in others it won't (run the migration, perform an insert to test).

If it fails, roll the migration back then modify it to read something like ...

```
CreateTable(
    "dbo.Foos",
    c => new
    {
        Id = c.Guid(nullable: false, defaultValueSql: "newid()"),
        ...
    })
.PrimaryKey(t => t.Id)
...;
```

... the extra code here tells SQL to generate the key as we would expect.

As a rule of thumb I would apply this change all the time for consistency reasons and it means that at a glance your migrations will show you exactly what keys are db generated.

edited May 31 at 17:17

answered May 1 at 11:47



War

6,347

2

36

82

## Entity Framework Core Update:

-1

There is no need to use `[DatabaseGenerated(DatabaseGeneratedOption.Identity)]` .  
There is no need to use `fluent API`

EF Core automatically take care of it and generates Id for primary key

Example:

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

Sign up with email



Sign up with Google

Sign up with Facebook



```
}
```

## Seeding

```
_context.DummyEntities.Add(new DummyEntity
{
    FirstName = "Abc",
    LastName = "Def",
    Postion = "User",
});

_context.SaveChanges();
```

answered Sep 2 '17 at 7:57



[immirza](#)

2,703 5 28 57

- 
- 1 On EntityFrameworkCore 1.0.4 this wasn't enough. I had to use `entity.Property(e => e.Id).ValueGeneratedOnAdd();` which did the trick for me. – [Martin Kunc](#) Sep 6 '17 at 6:30
- 
- 1 Down-voted because it does not appear this is accurate. When I generate my migration, the `defaultValue` is set to `new Guid("00000000-0000-0000-0000-000000000000")`). Please update your answer. – [Rogala](#) Oct 31 '17 at 18:03
- 
- 1 You're right that ef core takes care of it but it doesn't build a database table that properly takes care of auto generating that guid, for example in SQL server it doesn't setup a default or enable the auto generating property. – [Bailey Miller](#) Apr 30 at 18:18
- 
- 1 This answer is both wrong and extremely bad practice. there's so much wrong with this I would need a complete forum thread to correct all the issues in it. – [War](#) May 1 at 11:48
- 

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

Sign up with email



Sign up with Google

Sign up with Facebook

