How to re-create database for Entity Framework?

135

I have got into a bad state with my ASP.Net MVC 5 project, using Code-First Entity Framework. I don't care about losing data, I just want to be able to start fresh, recreate the database and start using Code-First migrations.

Currently I am in a state where every attempt to Update-Database results in an exception being thrown or getting an error message. Also the website can't access the database correctly. How can I wipe all migrations, re-create the database and start from scratch without having to create a new project? In other words, I want to keep my code but drop the database.

★ 69

Later I will also want to get the deployment database (SQL Server on Azure) in sync. Again, I don't mind dropping all the data - I just want to get it working.

Please provide any how-to steps to get back to a clean state. Much appreciated.

asp.net-mvc entity-framework azure azure-sql-database

edited Nov 1 '18 at 5:17
Pat Myron
702 1 11 28

asked Nov 30 '13 at 18:34
Toby Sharp
1,000 3 12 21

TBH if you simple want to `DROP DATABASE` then.... – Worthy7 Jul 15 '16 at 8:29

7 Answers

201

Follow below steps:

1) First go to Server Explorer in Visual Studio, check if the ".mdf" Data Connections for this project are connected, if so, right click and delete.

4) Delete Migrations folder by right click and delete.

5) Go to SQL Server Management Studio, make sure the DB for this project is not there, otherwise delete it.

6) Go to Package Manager Console in Visual Studio and type:

1. `Enable-Migrations -Force`
2. `Add-Migration init`
3. `Update-Database`

7) Run your application

Note: In step 6 part 3, if you get an error "Cannot attach the file...", it is possibly because you didn't delete the database files completely in SQL Server.

edited Jan 19 '16 at 13:59
Jess
13.7k 10 86 120

answered Nov 30 '13 at 18:57
Lin
13.3k 4 40 47

Excellent! This worked, thank you! Very helpful indeed. Are the steps similar for re-creating the deployment database (Azure SQL)? – Toby Sharp Dec 1 '13 at 16:01

hi @TobySharp, no problems. If you don't have any data in your database, you can follow these steps, but if you do have data, the steps are not quite the same. – Lin Dec 1 '13 at 16:30

3 Update, if you delete from VS (as the instructions say) and not Windows Explorer, then it works. Also, if you don't have SQL Management Studio, you can use Sql Object Explorer in VS to delete from a localdb. – Mike Ward Feb 24 '14 at 21:20

3 If you are getting the "Cannot attach file..." error, even though you believe you've deleted everything, try the answer from this thread: stackoverflow.com/questions/13275054/… – Andarta Jul 2 '15 at 0:02

1 @aggie, take a look at this link: goo.gl/3P65vu , hope it helps. – Lin Nov 20 '15 at 14:16

I would like to add that Lin's answer is correct.

a new mdf. The mdf does not have to match your old tables, because were going to delete it. So create or copy an old one to the correct folder.

2. Open it in server explorer [double click the mdf from solution explorer]
3. Delete it in server explorer
4. Delete it from solution explorer
5. run `update-database -force` [Use force if necessary]

Done, enjoy your new db

UPDATE 11/12/14 - I use this all the time when I make a breaking db change. I found this is a great way to roll back your migrations to the original db:

- Puts the db back to original
- Run the normal migration to put it back to current
    1. `Update-Database -TargetMigration:0 -force` [This will destroy all tables and all data.]
    2. `Update-Database -force` [use force if necessary]

edited Aug 11 '15 at 20:51
Oliver
7,016 4 52 82

answered Mar 25 '14 at 19:45
Steve Coleman
1,212 1 11 24

1 Great glad to help... I am going to update my answer. I recently found a better way to rollback all changes. – Steve Coleman Nov 12 '14 at 19:14

7

This worked for me:

1. Delete database from SQL Server Object Explorer in Visual Studio. Right-click and select delete.
2. Delete mdf and ldf files from file system - if they are still there.
3. Rebuild Solution.
4. Start Application - database will be re-created

1 Yeah, all I had to do was delete the MDF, update the connection string in the web.config, then run the application and register a new account. It created all of the tables again in the new database automagically. – Dan Bechard Dec 29 '16 at 21:15

While this question is premised by not caring about the data, sometimes maintenance of the data is essential.

2 If so, I wrote a list of steps on how to recover from Entity Framework nightmare when the database already has tables with the same name here: How to recover from Entity Framework nightmare - database already has tables with the same name

Apparently... a moderator saw fit to delete my post so I'll paste it here:

How to recover from Entity Framework nightmare - database already has tables with the same name

Description: If you're like us when your team is new to EF, you'll end up in a state where you either can't create a new local database or you can't apply updates to your production database. You want to get back to a clean EF environment and then stick to basics, but you can't. If you get it working for production, you can't create a local db, and if you get it working for local, your production server gets out of sync. And finally, you don't want to delete any production server data.

Symptom: Can't run ***Update-Database*** because it's trying to run the creation script and the database already has tables with the same name.

> Error Message: System.Data.SqlClient.SqlException (0x80131904): There is already an object named '' in the database.

Problem Background: EF understands where the current database is at compared to where the code is at based on a table in the database called dbo.__MigrationHistory. When it looks at the Migration Scripts, it tries to reconsile where it was last at with the scripts. If it can't, it just tries to apply them in order. This means, it goes back to the initial creation script and if you look at the very first part in the UP command, it'll be the CreeateTable for the table that the error was occurring on.

To understand this in more detail, I'd recommend watching both videos referenced here: https://msdn.microsoft.com/en-us/library/dn481501(v=vs.113).aspx

Solution: What we need to do is to trick EF into thinking that the current database is up to date while not applying these CreateTable commands. At the same time, we still want those commands to exist so we can create new local databases.

Step 2: Local environment clean Open your migrations folder and delete it. I'm assuming you can get this all back from git if necessary.

Step 3: Recreate Initial In the Package Manager, run "Enable-Migrations" (EF will prompt you to use -ContextTypeName if you have multiple contexts). Run "Add-Migration Initial -verbose". This will Create the initial script to create the database from scratch based on the current code. If you had any seed operations in the previous Configuration.cs, then copy that across.

Step 4: Trick EF At this point, if we ran ***Update-Database***, we'd be getting the original error. So, we need to trick EF into thinking that it's up to date, without running these commands. So, go into the Up method in the Initial migration you just created and comment it all out.

Step 5: Update-Database With no code to execute on the Up process, EF will create the dbo.__MigrationHistory table with the correct entry to say that it ran this script correctly. Go and check it out if you like. Now, uncomment that code and save. You can run ***Update-Database*** again if you want to check that EF thinks its up to date. It won't run the Up step with all of the CreateTable commands because it thinks it's already done this.

Step 6: Confirm EF is ACTUALLY up to date If you had code that hadn't yet had migrations applied to it, this is what I did...

Run "Add-Migration MissingMigrations" This will create practically an empty script. Because the code was there already, there was actually the correct commands to create these tables in the initial migration script, so I just cut the CreateTable and equivalent drop commands into the Up and Down methods.

Now, run ***Update-Database*** again and watch it execute your new migration script, creating the appropriate tables in the database.

Step 7: Re-confirm and commit. Build, test, run. Ensure that everything is running then commit the changes.

Step 8: Let the rest of your team know how to proceed. When the next person updates, EF won't know what hit it given that the scripts it had run before don't exist. But, assuming that local databases can be blown away and re-created, this is all good. They will need to drop their local database and add create it from EF again. If they had local changes and pending migrations, I'd recommend they create their DB again on master, switch to their feature branch and re-create those migration scripts from scratch.

edited Sep 7 '17 at 9:49

answered Sep 7 '17 at 3:34
System.Exception
114 7

list.

answered May 25 '15 at 10:13
Dudi
2,335 1 18 22

If you want to add something to an answer, please add a comment to that answer rather than posting a new answer. – Oliver Aug 11 '15 at 11:04

You right but I was unable to comment in that time. Today I can.. – Dudi Aug 11 '15 at 21:44

1 I wasn't aware of that. So now you can move your answer into a comment and delete this answer since technically it is not an (complete) answer to the question :-) – Oliver Aug 12 '15 at 7:48

0

A possible very simple fix that worked for me. After deleting any database references and connections you find in server/serverobject explorer, right click the App_Data folder (didn't show any objects within the application for me) and select open. Once open put all the database/etc. files in a backup folder or if you have the guts just delete them. Run your application and it should recreate everything from scratch.

answered Jul 31 '14 at 18:47
edencorbin
1,101 14 32

0

My solution is best suited for :
- deleted your mdf file
- want to re-create your db.

In order to **recreate your database** you need add the connection using Visual Studio.

Step 1 : Go to Server Explorer add new connection(or look for a add db icon).

AttachDbFilename attribute)

Step 4 : click browse and navigate to where you will like it to be located.

Step 5 : in the package manager console run command `update-database`

answered Oct 2 '17 at 11:52

Mbuso Mkhize
81 1 4