Update-Database command is not working in ASP.Net Core / Entity Framework 6 because object in database already exists

2

1

I was updating my database by the command line, but then I manually updated one of my tables.

This seems to have disrupted my ability to update-database. I receive the following error when I try to update:

```
 System.Data.SqlClient.SqlException: There is already an object named
'ClientsAndTestimonials' in the database.    at
System.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean
breakConnection, Action`1 wrapCloseInAction)
   at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject
stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)
    at System.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand
cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler,
TdsParserStateObject stateObj, Boolean& dataReady)
   at System.Data.SqlClient.SqlCommand.RunExecuteNonQueryTds(String methodName, Boolean
async, Int32 timeout, Boolean asyncWrite)
    at System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery(TaskCompletionSource`1
completion, Boolean sendToPipe, Int32 timeout, Boolean asyncWrite, String methodName)
    at System.Data.SqlClient.SqlCommand.ExecuteNonQuery()
    at
Microsoft.EntityFrameworkCore.Storage.Internal.RelationalCommand.Execute(IRelationalConnec
 connection, String executeMethod, IReadOnlyDictionary`2 parameterValues, Boolean
openConnection, Boolean closeConnection)
    at
Microsoft.EntityFrameworkCore.Storage.Internal.RelationalCommand.ExecuteNonQuery(IRelation
 connection, IReadOnlyDictionary`2 parameterValues, Boolean manageConnection)
    at
Microsoft.EntityFrameworkCore.Migrations.Internal.MigrationCommandExecutor.ExecuteNonQuery
 migrationCommands, IRelationalConnection connection)
    at Microsoft.EntityFrameworkCore.Migrations.Internal.Migrator.Migrate(String
targetMigration)
    at Microsoft.EntityFrameworkCore.Design.MigrationsOperations.UpdateDatabase(String
targetMigration, String contextType)
    at Microsoft.EntityFrameworkCore.Tools.Cli.DatabaseUpdateCommand.
```

```
There is already an object named 'ClientsAndTestimonials' in the database.
```

I have been trying to fix this problem for the past few days. Most fellow developers suggest some variation of using Add-migration "Reset" -IgnoreChanges, like John Salewski from the following link.

However, I keep getting an error that says "A parameter cannot be found that matches parameter name 'IgnoreChanges'".

Any suggestions would be greatly appreciated!

c# sql database asp.net-core entity-framework-6

edited May 23 '17 at 12:02
Community ♦
1 1

asked Apr 28 '17 at 19:17
Kelsey Steele
34 1 5

Try -Ignore_Changes – WEI_DBA Apr 28 '17 at 19:20

do you have any Data in the DB that you need, if not just drop the whole db and re add it – johnny 5 Apr 28 '17 at 19:30

@WEI_DBA, that didn't work, I got the same error. Any other ideas? – Kelsey Steele Apr 28 '17 at 19:31

@johnny5 I do have a lot of Data in the DB. Is there anything else I can do, that lets me keep the data? – Kelsey Steele Apr 28 '17 at 19:34

@KelseySteele I believe Steve Greene's answer should work – johnny 5 Apr 28 '17 at 19:41

1 Answer

First, that is not EF 6, it is EF Core. There is no -IgnoreChanges currently in EF Core (see here) but you can achieve the equivalent by commenting out all the code in the Up() method and applying the migration. This will take a snapshot of the current model state so that subsequent migrations will only include changes from that point forward.

Steve Greene
9,656 1 22 39

I am still getting the same error. I'm not sure if my sequence is correct. The Up() method is within the migration. So I have to first create a new migration to see the Up() method, but I think this is creating the snapshot or at least I think so because it's called ApplicationDbContextModelSnapshot. I can't see how I can comment out the Up() method without creating a snapshot that includes subsequent migrations. Should I delete all of my previous migrations? – Kelsey Steele Apr 28 '17 at 20:01

2 OK, didn't realize you had prior migrations. So it sounds like you added ClientsAndTestimonials to your database manually instead of letting migrations create it, so now when you add that to your models EF tries to add it again. Same concept applies - look at your Up() code and comment out the lines that create objects that already exist. You can always reset migrations as long as all the deployed databases are up to that point. – Steve Greene Apr 29 '17 at 13:58

Thank you, Steve! That seemed to do the trick. I followed the directions from your link to reset migrations and now I can update-database. I really appreciate your help. – Kelsey Steele May 1 '17 at 15:29

should i comment my very first (items inside) up & down migration in order to have base model snapshot? – ManirajSS Sep 28 '17 at 10:50

1 Yes. The model stored in resource field in code is the key thing. It is used to do the next compare (the database record is not). Plus, if you start a new database it will still create all your objects because it would be comparing an empty model to the stored model. – Steve Greene Sep 28 '17 at 14:00