Testing EF CORE

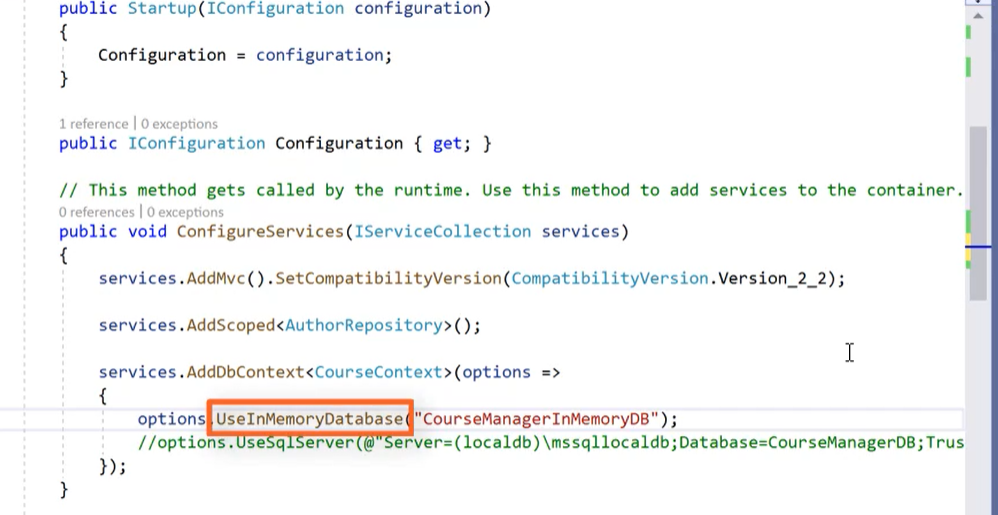
Spis treści

[InMemory Tests 2](#_Toc63522179)

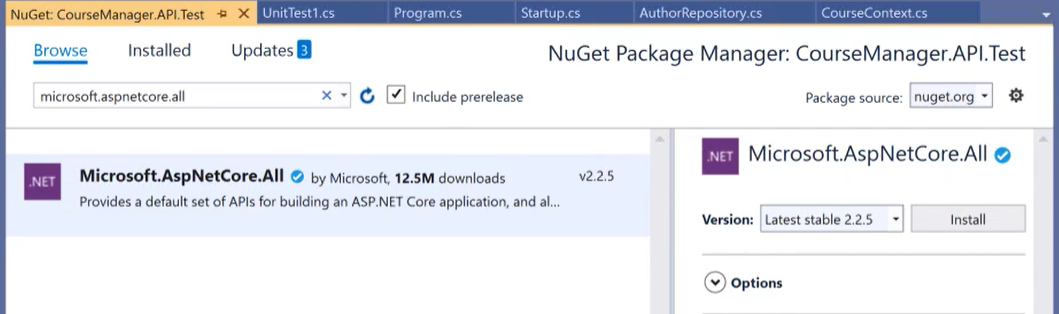
[My own Implementation 5](#_Toc63522180)

[SQLLite Tests 6](#_Toc63522181)

# InMemory Tests



To use memory db required is inject option in Startup class in configuration services metod. Option should use memory database.



To make unit tests in tests project required is package from nugget.



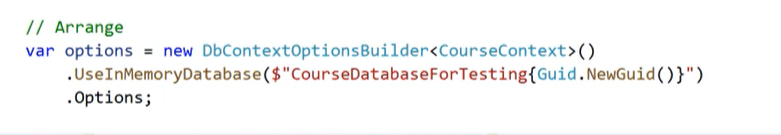
Patter for inMemory test. Required are injected option with use in memory database(“dbName”);



After all configuration dummy data has to be save.



After set up, InMemory testing option is available.



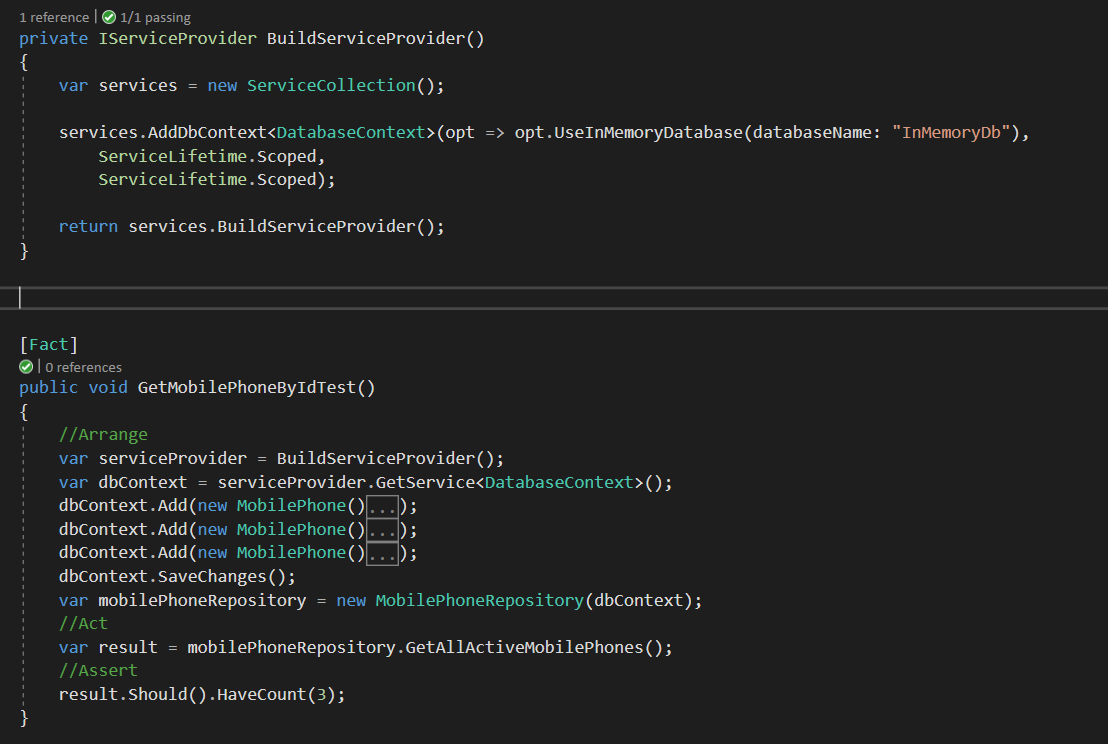
To isolate test from other unit test use Guid to create unique inMemory DB



Tests can be isolated by create new context in another using clause.

But the Biggest problem with inMemory DB is that this Type of DB is not relational if entity has foreign key which is required then Db doesn’t crush so in UT you don’t check all relation or if data if retrieve in good way. In this approach can be test only crud method and it’s working in good way or not.

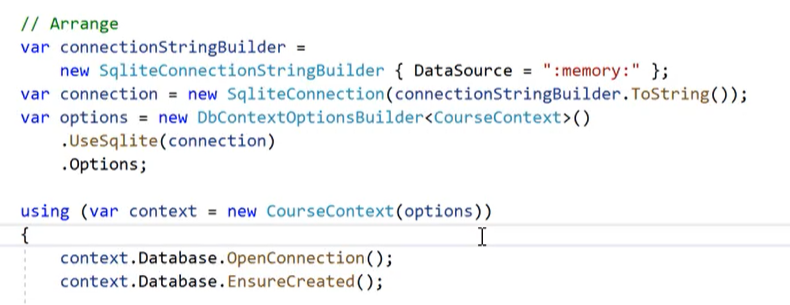
## My own Implementation



# SQLLite Tests

Remember about install Microsoft.AspNetCore.all

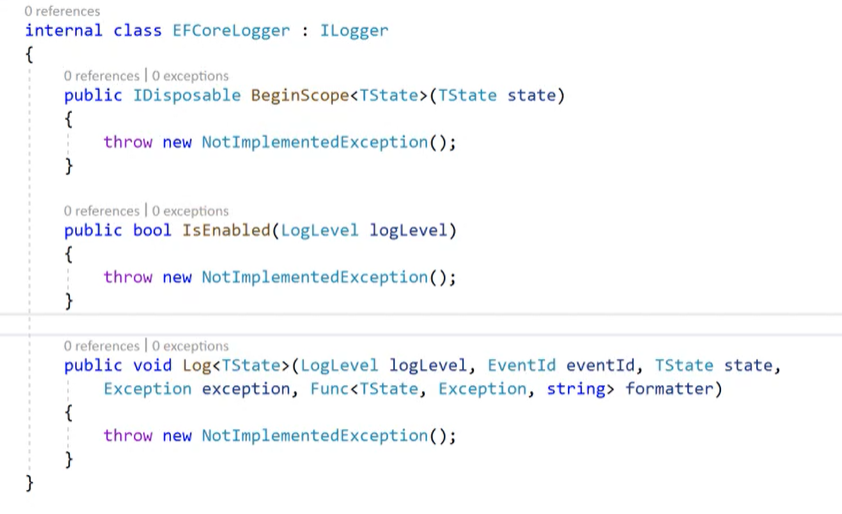
SQLLite is much closer to normal db, but all data is in memory so it’s useful to make tests for EF.



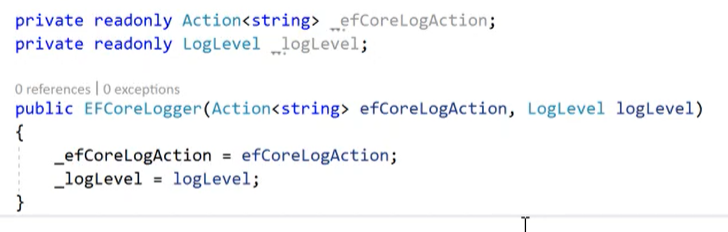
Sql lite configuration is similar to inmemory configuration. SqlLite connection require connefction string and in DbContext has to be invoke UseSQLite(connection). After this step in using claus DbContext need has open connection and created db. When this stem is accomplished unit test can be made.

When context has open connection, it will be open in each context when all test will be finished.

When test failed there is not information about it. To display that info needed is a Logger.



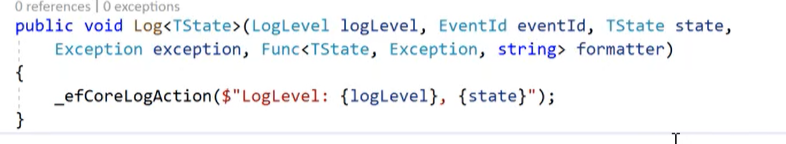
EFCoreLogger inherit method from ILogger.



Logger required two properties



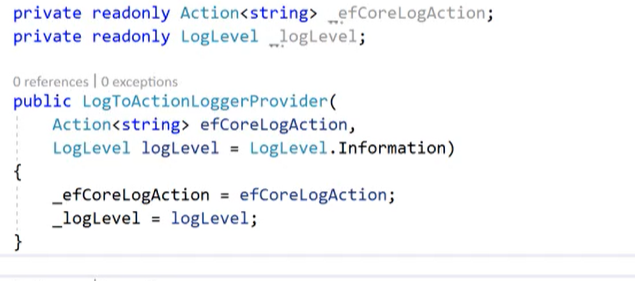
Implementation of first two method is easy.

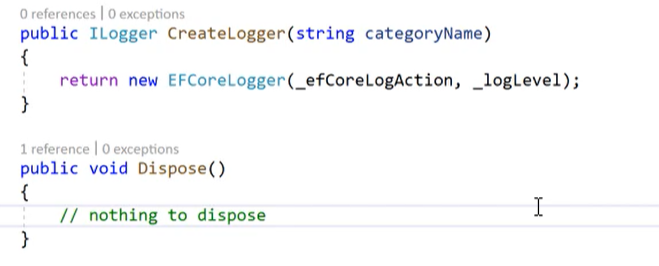


In third one required is efCoreAction with string as parameter



When logger is configured. Required is a LoggerProvider with implemented interfaces with the same name.

The same like in Logger class are required and has to be initialize in constructor.

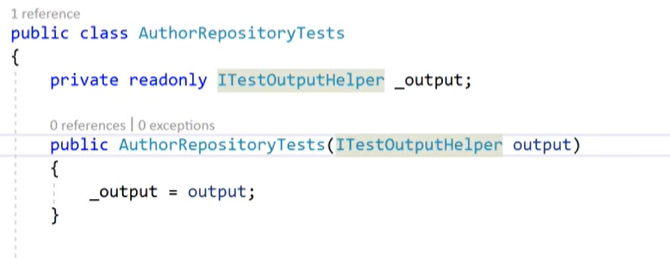


Create method return logger and Dispose do nothing.

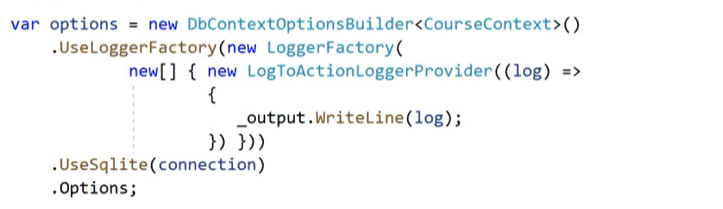


Now it’s time to add logger to DBcollection Context. Logger needs collection of string to this object will be add logs. Before useSqlite method must be invoked method UseLoggerFActor with our logger passed inside anonymous object with lambda expression. After all this operations logger will save all log to collection when all will be good or when any problem occur.

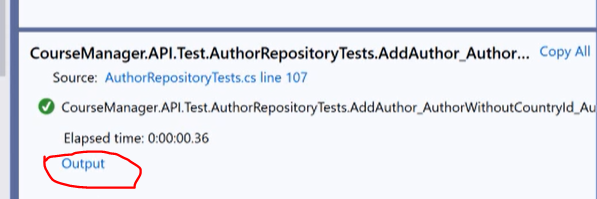
Another option is add logger to Explorer window and read all logs.



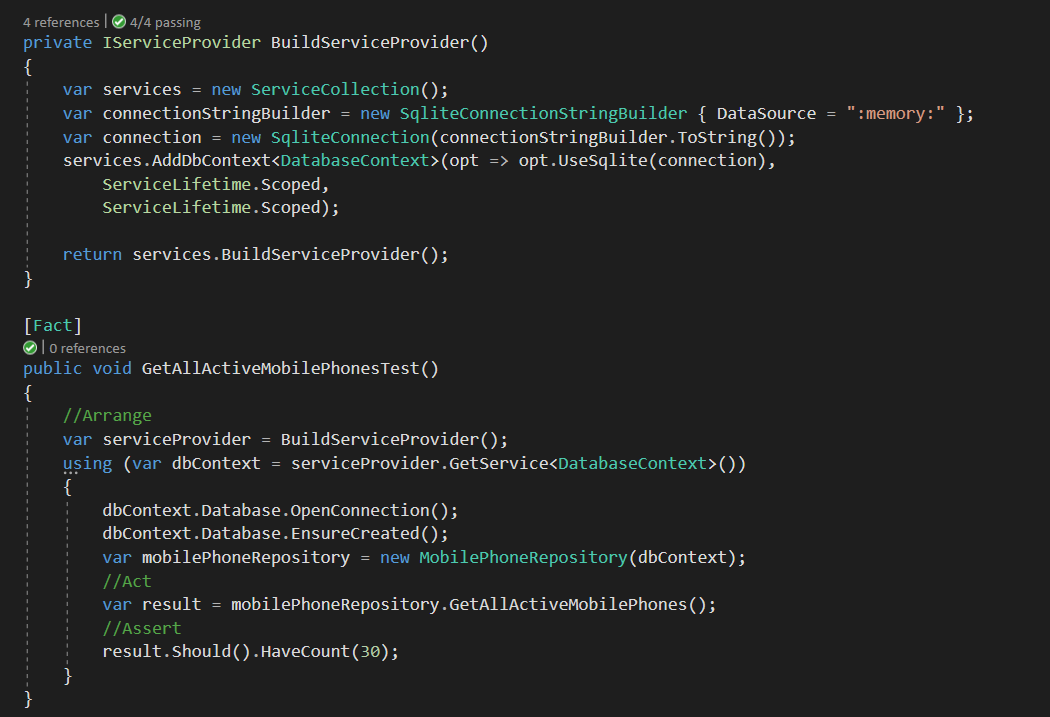
Test class required ITEstOutputHelper. This object must be initialize in constructor. After this step…

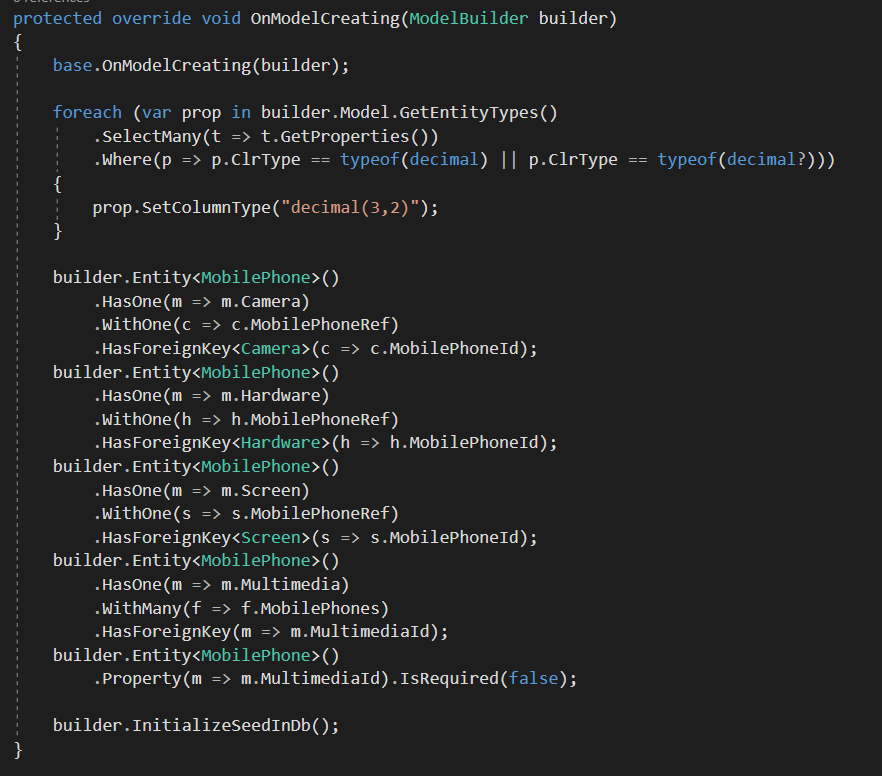


Instead of collection of string inside lambda use is an output object with method WriteLine and log as parameter. This method display all logs in Explorer windows. Click on output in test method.



## My implementation





Sqlite db has data and relation like normal Db used in app.