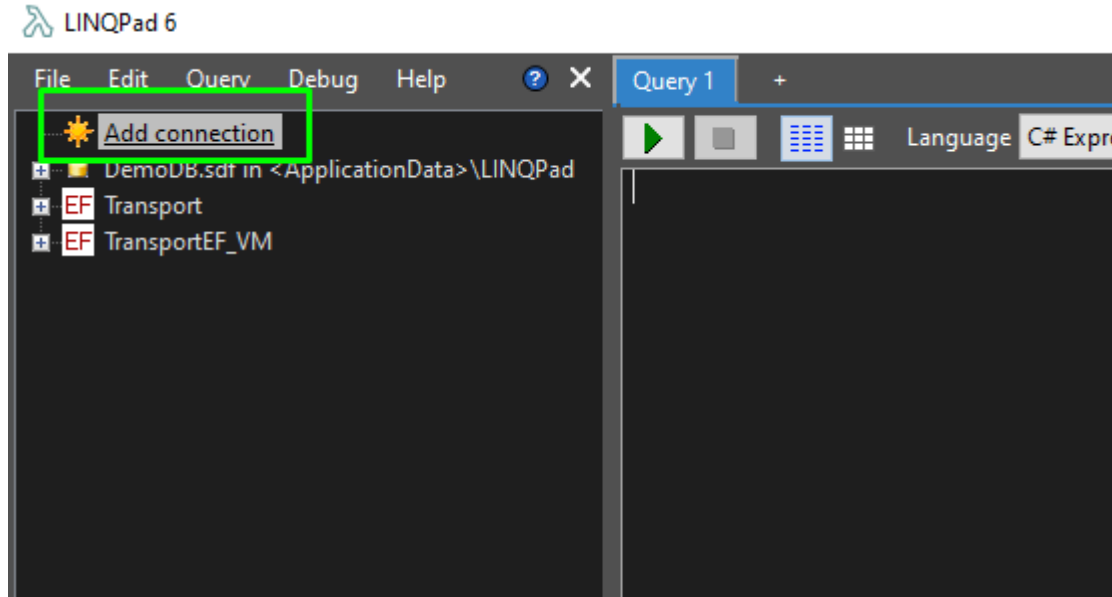


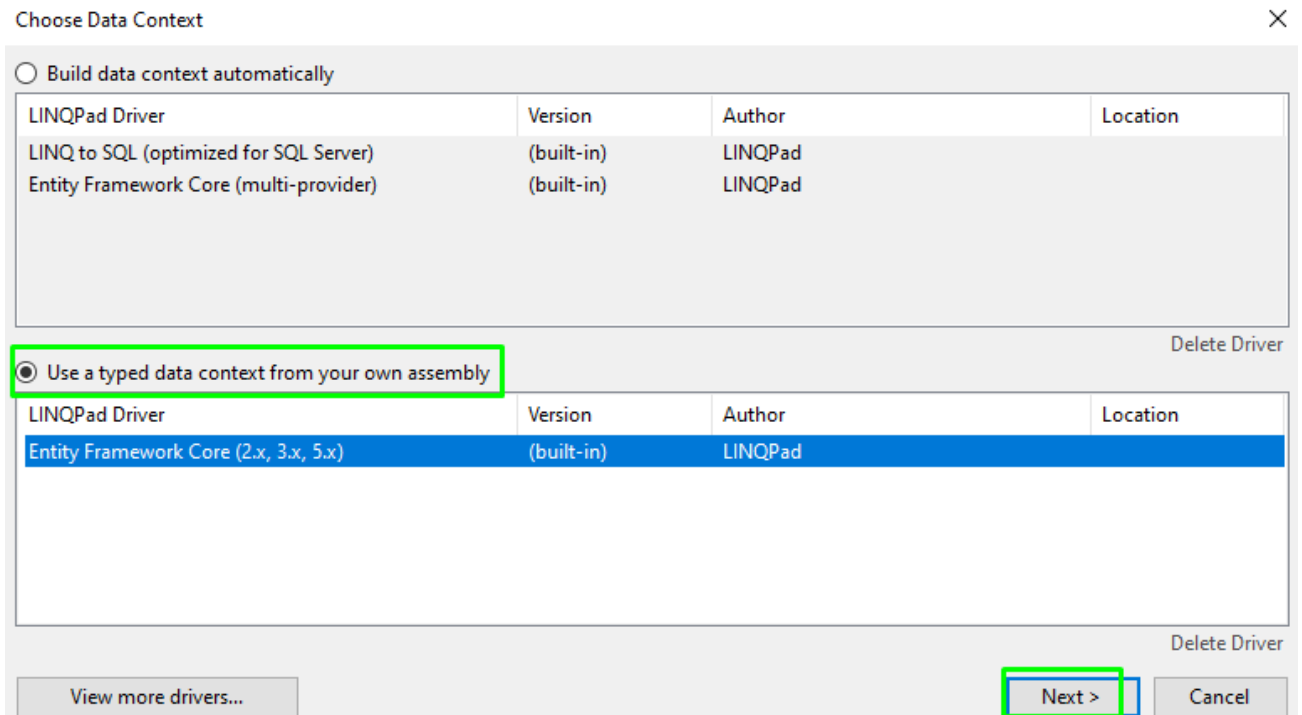
Тестирование запросов через LINQPad

Запросы, выполняемые в приложении через Entity Framework, можно тестировать при помощи утилиты *LINQPad*.

1. Скачать дистрибутив **LINQPad 6**
<https://www.linqpad.net/>
2. Установить
3. Запустить
4. Нажать **Add connection**

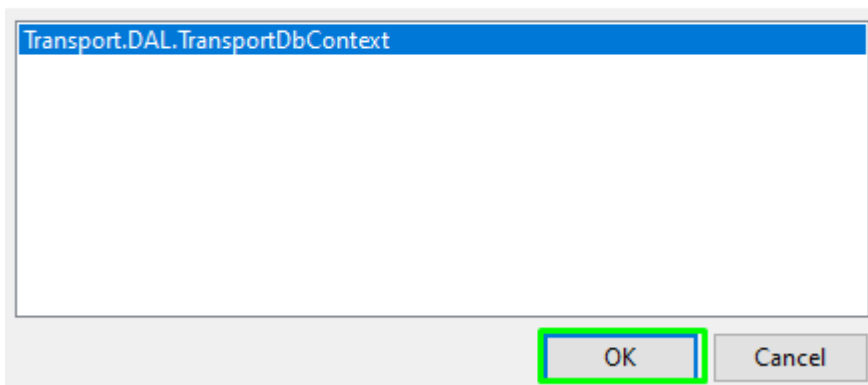


5. В появившемся окне настроить подключение

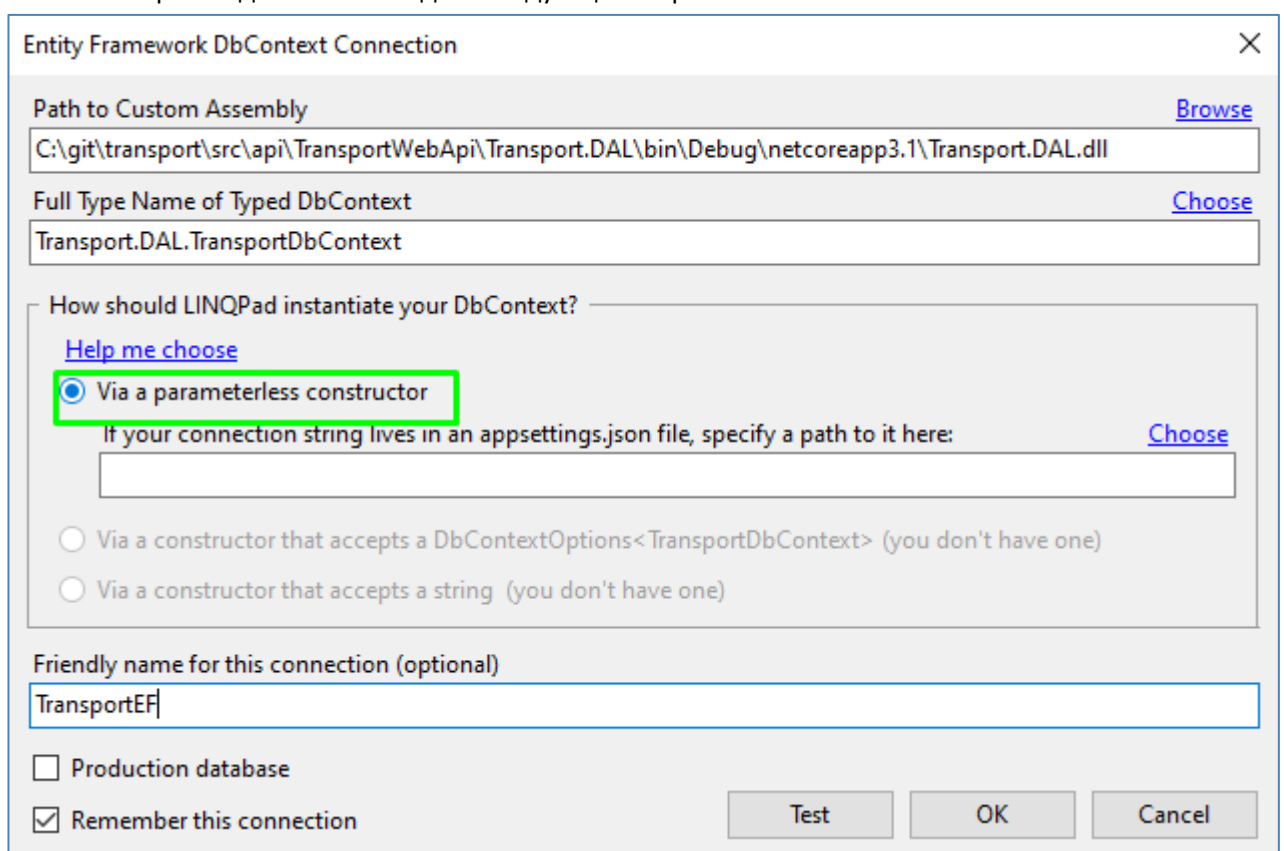


Предварительно нужно скомпилировать бэкенд в Visual Studio, т.к. далее нужно будет указать путь к Transport.DAL.dll. После указания пути к DLL, в следующем окне нажать **OK**.

C:\git\transport\src\api\TransportWebApi\Transport.DAL\bin\Debug\netcoreap...

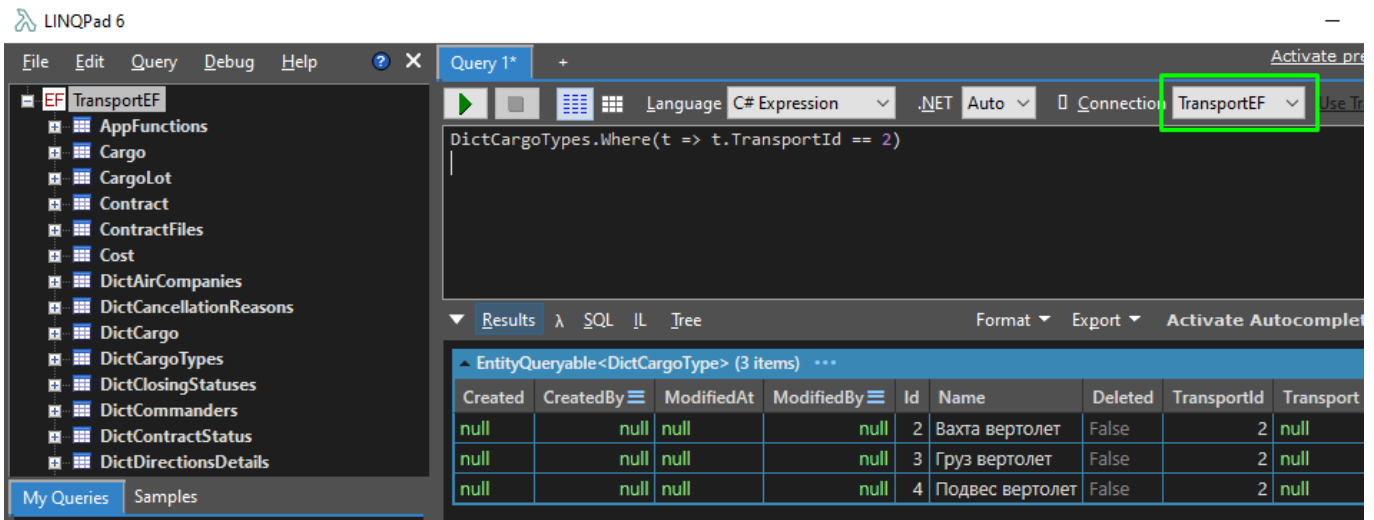


В итоге настройки должны выглядеть следующим образом:

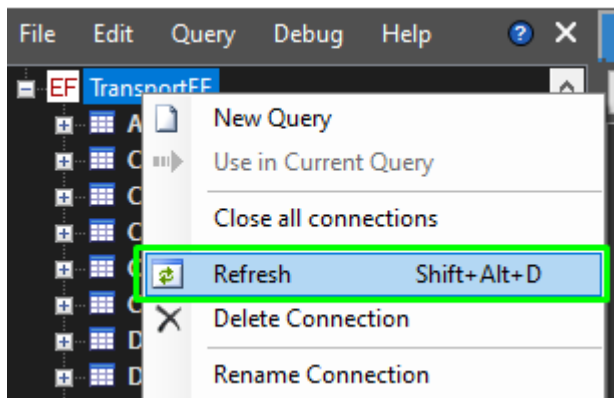


Подключение будет выполняться к тестовой БД.

6. После успешной настройки подключения слева в дереве должны отобразиться DbSet'ы, а справа в редакторе можно писать запросы к БД (таким же образом, как в коде приложения).



При изменении модели (т.е. структуры БД) нужно перекомпилировать бэкенд, а в LINQPad через контекстное меню подключения выполнить **Refresh**.



Пример. Протестируем метод `GetGeneralOrders` из `OrderFactRepository`.

```
public Task<List<OrderFactDTO>> GetGeneralOrders(DateTime dateDeparture, int transportId, Cancell
{
    var query = from order in Context.OrderFact
        where order.TransportId == transportId &&
            order.DateDeparture.Date >= dateDeparture.Date.AddDays(-4) &&
            order.DateDeparture.Date <= dateDeparture.Date.AddDays(4)
        orderby order.OrderNumber
        select new OrderFactDTO
        {
            Id = order.Id,
            Name = order.OrderNumber
        };

    return query.AsNoTracking().ToListAsync(cancellation token);
}
```

Скопируем запрос в LINQPad, в поле **Language** выберем C# Statement(s), инициализируем параметры для запроса и выполним его (**F5**):

Query 1* + [Activate premium](#)

Language: C# Statement(s) .NET Auto Connection TransportEF

```
var Context = this;

var transportId = 2;
var dateDeparture = new DateTime(2021, 4, 11);

var query = from order in Context.OrderFact
            where order.TransportId == transportId &&
                  order.DateDeparture.Date >= dateDeparture.Date.AddDays(-4) &&
                  order.DateDeparture.Date <= dateDeparture.Date.AddDays(4)
            orderby order.OrderNumber
            select new /*OrderFactDTO*/
            {
                Id = order.Id,
                Name = order.OrderNumber
            };

query.Dump();
```

Results SQL IL Tree Format Export [Activate Autocompletion](#)

EntityQueryable<> (92 items) ...

Id	Name
53737	A-100-21-B
53741	A-103-21-B
53795	A-117-21-B
53813	A-119-21-B
53900	A-122-21-B